# Functions (4A)

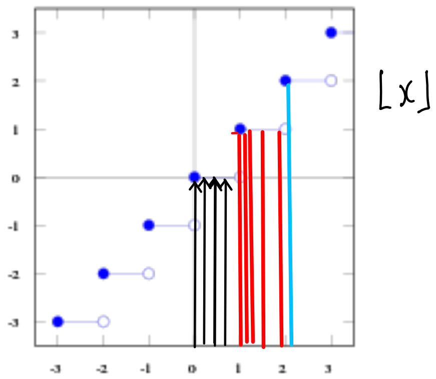
Young Won Lim
4/13/18

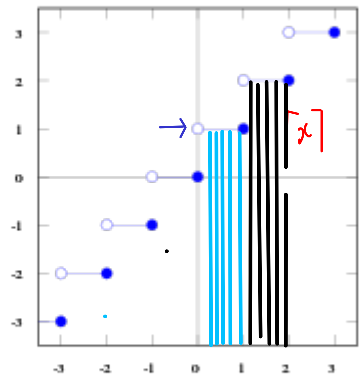
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

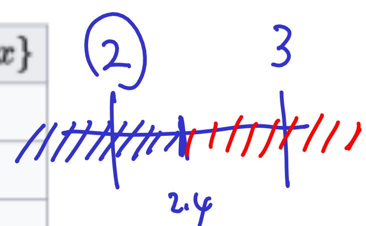
# Floor and Ceiling Functions

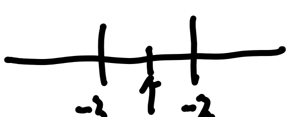
floor $\lfloor x \rfloor$ integer    ceiling $\lceil x \rceil$ integer

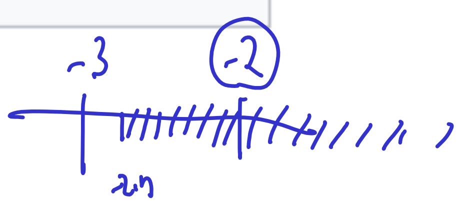



| x | Floor $\lfloor x \rfloor$ | Ceiling $\lceil x \rceil$ | Fractional part $\{x\}$ |
|---|---|---|---|
| **2** | 2 | 2 | 0 |
| **2.4** | 2 | 3 | 0.4 |
| **2.9** | 2 | 3 | 0.9 |
| **−2.7** | −3 | −2 | 0.3 |
| **−2** | −2 | −2 | 0 |

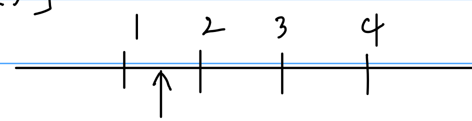


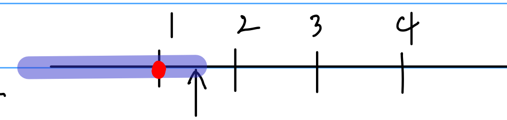




https://en.wikipedia.org/wiki/Function_(mathematics)

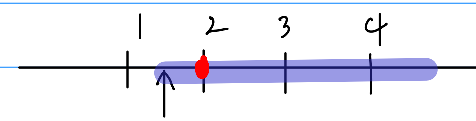
$x = \sqrt{2} \approx 1.414$ $\lfloor x \rfloor$




floor $\lfloor x \rfloor = 1$




ceiling $\lceil x \rceil = 2$



# Floor and Ceiling Functions

In the following formulas, $x$ and $y$ are real numbers, $k$, $m$, and $n$ are integers, and $\mathbb{Z}$ is the set of integers (positive, negative, and zero).

Floor and ceiling may be defined by the set equations

floor $\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \le x\},$

ceiling $\lceil x \rceil = \min\{n \in \mathbb{Z} \mid n \ge x\}.$

Since there is exactly one integer in a half-open interval of length one, for any real $x$ there are unique integers $m$ and $n$ satisfying

$$x - 1 < m \le x \le n < x + 1.$$

Then $\lfloor x \rfloor = m$ and $\lceil x \rceil = n$ may also be taken as the definition of floor and ceiling.

https://en.wikipedia.org/wiki/Function_(mathematics)

# Floor and Ceiling Functions

the floor function is the function that takes as input a real number x and gives as output the **greatest** integer less than or equal to x, denoted **floor(x) = $\lfloor x \rfloor$**.

Similarly, the ceiling function maps x to the **least** integer greater than or equal to x, denoted **ceiling(x) = $\lceil x \rceil$**.

https://en.wikipedia.org/wiki/Function_(mathematics)

## References


[1] http://en.wikipedia.org/
[2]

# Relations (3A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

# Transitive Relation

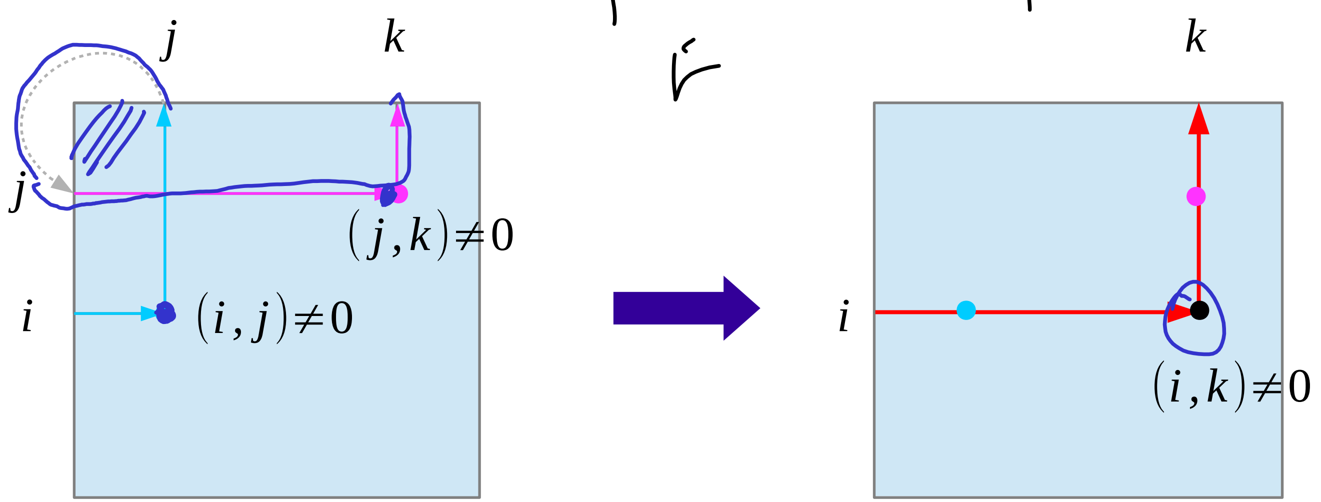
T → T  T
T → F  F
F → T  T
F → T  T

$$\forall x, \forall y, \forall z \ \left[ \left( (x, y) \in R \wedge (y, z) \in R \right) \rightarrow (x, z) \in R \right]$$

F

F

F




$(i \ R \ j) \wedge (j \ R \ k)$ $\quad\quad$ $(i \ R \ k)$

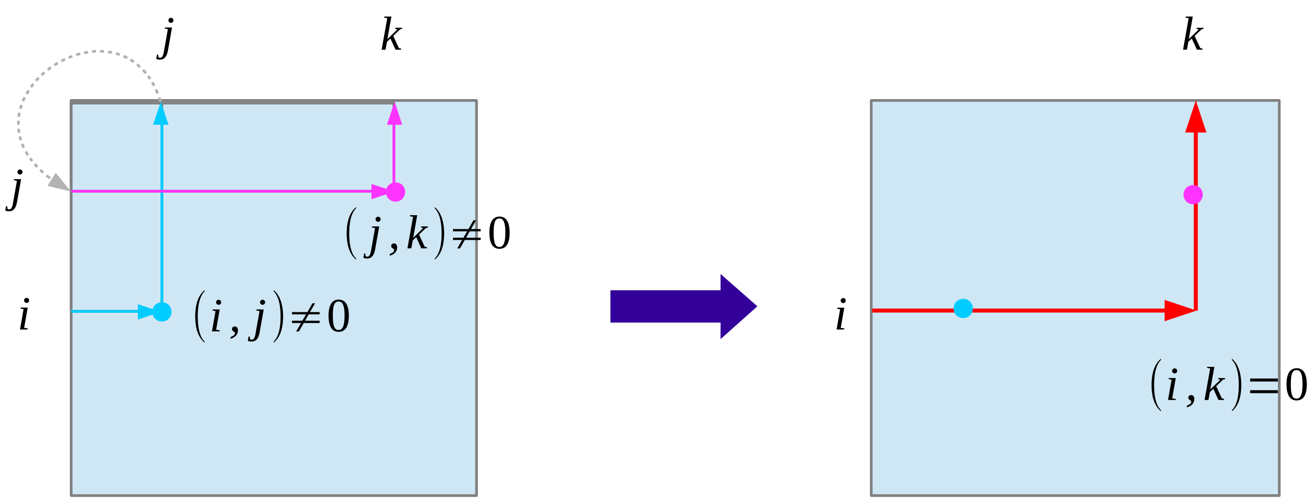
# Not Transitive Relation

$\neg\{\forall x, \forall y, \forall z\ [((x, y) \in R \wedge (y, z) \in R) \rightarrow (x, z) \in R\ ]\}$

$\exists x, \exists y, \exists z\ \neg[((x, y) \in R \wedge (y, z) \in R) \rightarrow (x, z) \in R\ ]$

$\exists x, \exists y, \exists z\ \neg[\neg((x, y) \in R \wedge (y, z) \in R) \vee ((x, z) \in R)\ ]$

$\exists x, \exists y, \exists z\ [(x, y) \in R \wedge (y, z) \in R \wedge ((x, z) \notin R)\ ]$




$(i\ R\ j) \wedge (j\ R\ k)$ $\quad\quad$ $(i\ \cancel{R}\ k)$

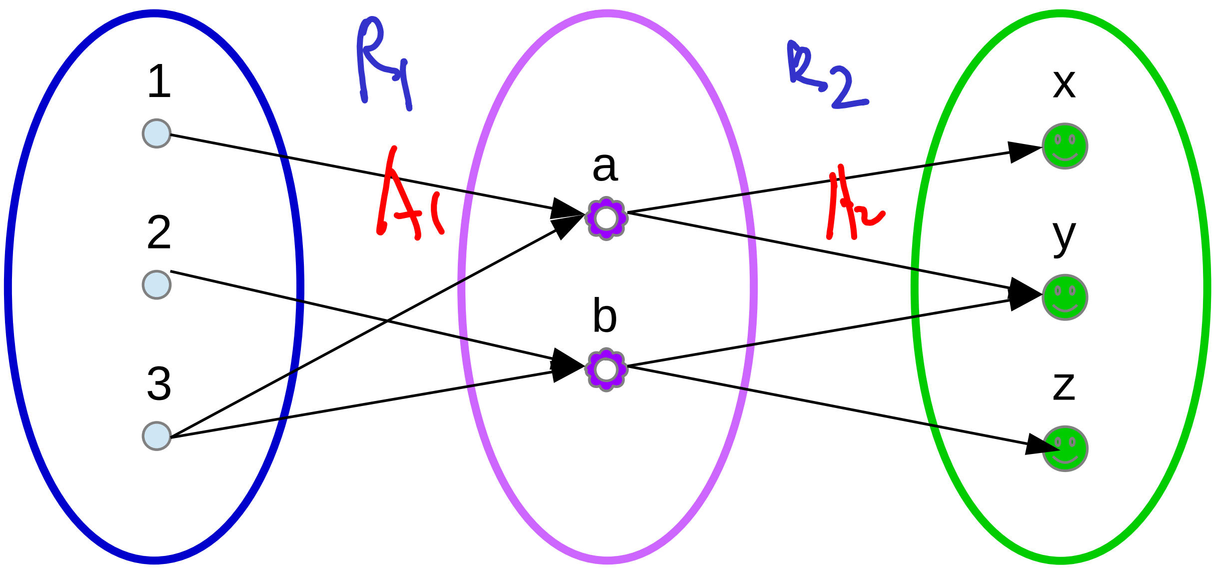
# Relation Examples

$$R_1 \in \{(1,a),\ (2,b),\ (3,a),\ (3,b)\}$$

$$R_2 \in \{(a,x),\ (a,y),\ (b,y),\ (b,z)\}$$

$R_2 \circ R_1$

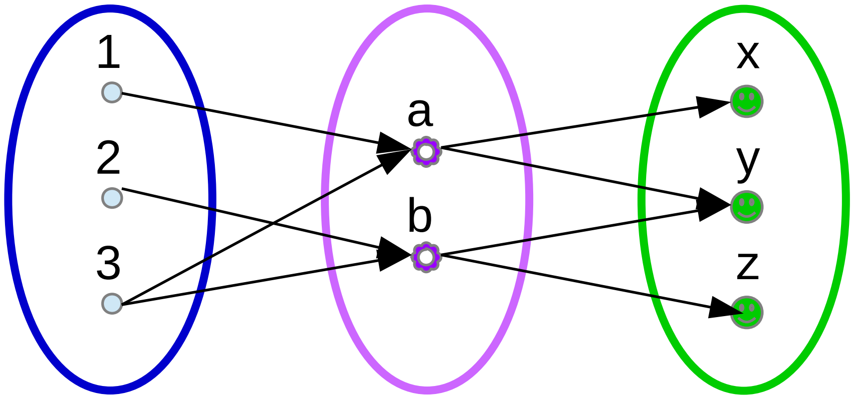
$A_1 A_2$

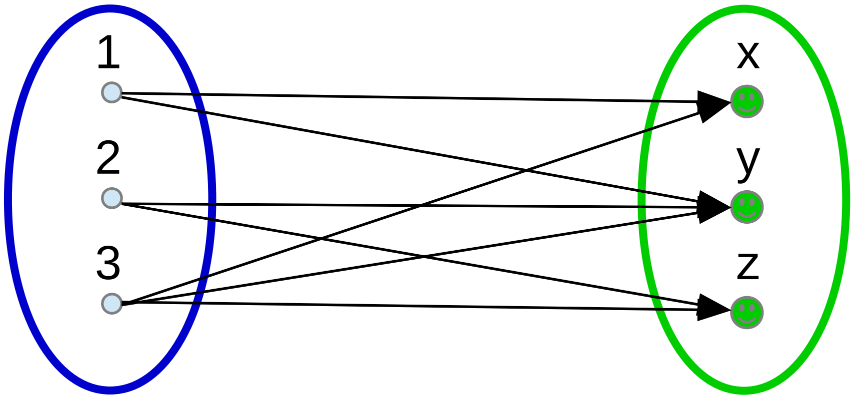


# Composite Relation Examples

$R_1 \in \{(1,a),\ (2,b),\ (3,a),\ (3,b)\}$

$R_2 \in \{(a,x),\ (a,y),\ (b,y),\ (b,z)\}$



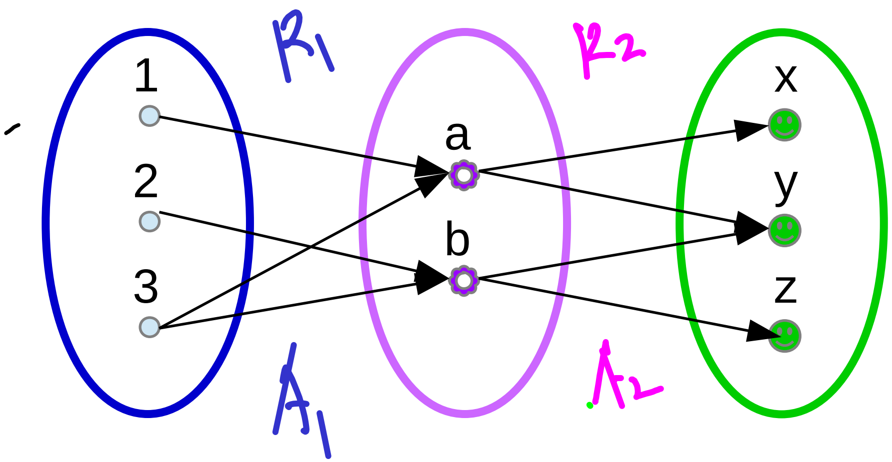

$R_2 \circ R_1 \in \{(1,x),\ (1,y),\ (2,y),\ (2,z),\ (3,x),\ (3,y),\ (3,z)\}$

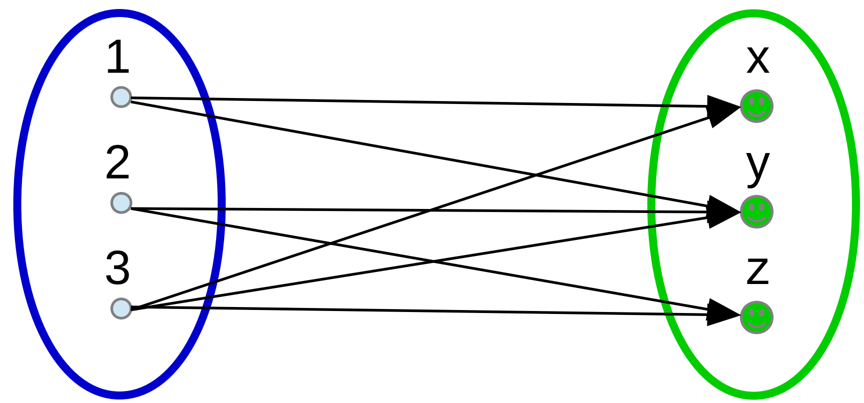
# Composite Relation Examples

$R_1 \in \{(1,a),\ (2,b),\ (3,a),\ (3,b)\}$

$R_2 \in \{(a,x),\ (a,y),\ (b,y),\ (b,z)\}$

$R_1 \circ R_2 \in \{(1,x),\ (1,y),\ (2,y),\ (2,z),\ (3,x),\ (3,y),\ (3,z)\}$




$$A_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \end{array} \qquad A_2 = \begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

$$A_1 A_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \end{array}$$

# Matrix of a Relation

$R_1 \in \{(1,a),\ (2,b),\ (3,a),\ (3,b)\}$

$$A_1 = \begin{array}{c c} & \begin{array}{cc} a & b \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \end{array}$$

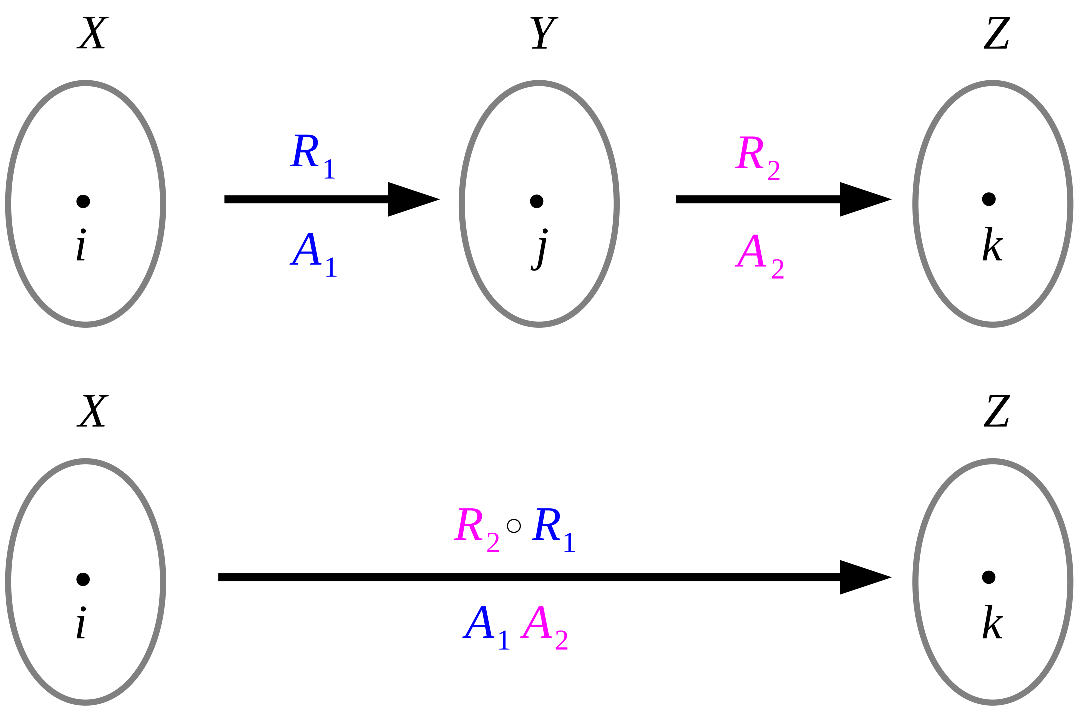
$R_2 \in \{(a,x),\ (a,y),\ (b,y),\ (b,z)\}$

$$A_2 = \begin{array}{c c} & \begin{array}{ccc} x & y & z \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

$R_2 \circ R_1 \in \{(1,x),\ (1,y),\ (2,y),\ (2,z),\ (3,x),\ (3,y),\ (3,z)\}$

$$A_1 A_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{array}{c c} & \begin{array}{ccc} x & y & z \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \end{array}$$

# Composite Relation Properties




$$(i, k) \in R_2 \circ R_1 \quad \Longleftrightarrow \quad a_{ik} \neq 0 \quad of \ A_1 A_2$$

# Composite Relation Examples

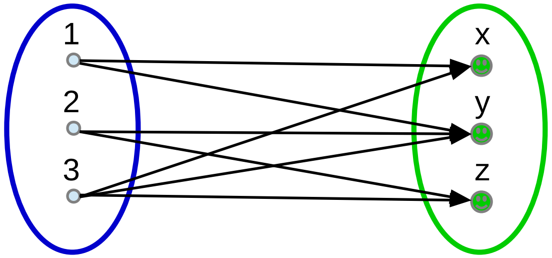
$R_1 \in \{(1,a),\ (2,b),\ (3,a),\ (3,b)\}$

$R_2 \in \{(a,x),\ (a,y),\ (b,y),\ (b,z)\}$

$R_2 \circ R_1 \in \{(1,x),\ (1,y),\ (2,y),\ (2,z),\ (3,x),\ (3,y),\ (3,z)\}$

$$A_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \end{array} \qquad A_2 = \begin{array}{c} \\ a \\ b \end{array}\begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array} \qquad A_1 A_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \end{array}$$

$\boxed{(i,k) \in R_2 \circ R_1} \iff \boxed{a_{ik} \neq 0 \quad of\ A_1 A_2}$



# Composite Relation Property Examples

$A_1$ $A_2$ 　　 $R_2 \circ R_1$

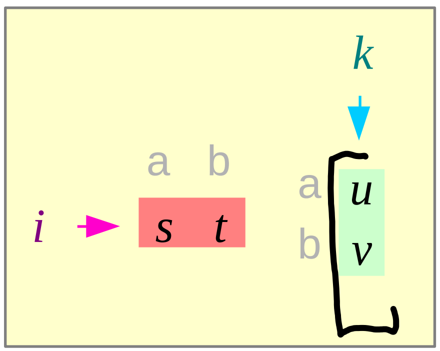
$$A_1 A_2 = \begin{bmatrix} 1 & 0 \\ s & t \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & u & 0 \\ 0 & v & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

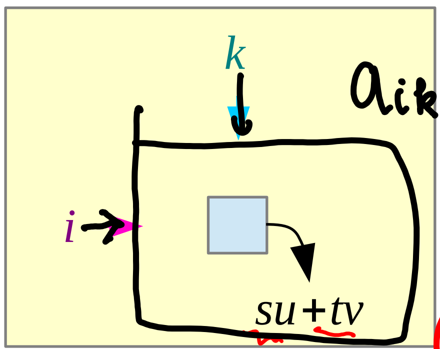
$A_1 \cdot A_2$

$2 \to 2$

$$R_1 \quad A_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \end{array}$$

$$R_2 \quad A_2 = \begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$




$a_{ik} \neq 0$

$a_{ik} \neq 0$

$i \in \{1, 2, 3\}$
$k \in \{x, y, z\}$

$s \in \{0, 1\}$
$t \in \{0, 1\}$
$u \in \{0, 1\}$
$v \in \{0, 1\}$

nonzero $(i,k)^{th}$ element of $A_1 A_2$ ⟺ $(i,k) \in R_2 \circ R_1$

$su + tv \neq 0$

$(su \neq 0) \lor (tv \neq 0)$

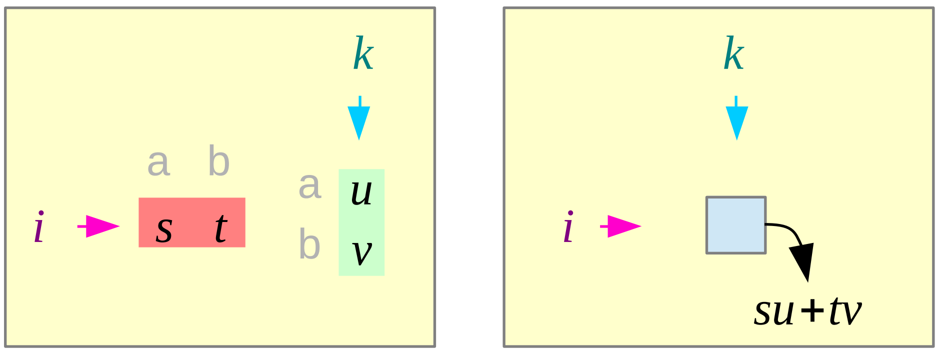
$(s=1 \land u=1) \lor (t=1 \land v=1)$

# Sufficient Part

$$A_1 A_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & \boxed{1} & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

$$A_1 = \begin{array}{c|cc} & a & b \\ \hline 1 & 1 & 0 \\ 2 & 0 & 1 \\ 3 & 1 & 1 \end{array}$$

$$A_2 = \begin{array}{c|ccc} & x & y & z \\ \hline a & 1 & 1 & 0 \\ b & 0 & 1 & 1 \end{array}$$




$i \in \{1, 2, 3\}$

$k \in \{x, y, z\}$

$s \in \{0, 1\}$

$t \in \{0, 1\}$

$u \in \{0, 1\}$

$v \in \{0, 1\}$

$\boxed{(a_{ik} \neq 0) \quad su+tv \neq 0}$ ⇒

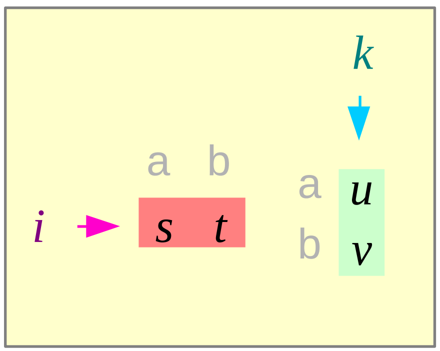
| $su = 1$ | $tv = 1$ |
|---|---|
| $(s = 1)$ $(u = 1)$ | $(t = 1)$ $(v = 1)$ |
| $(i, a) \in R_1$ $(a, k) \in R_2$ | $(i, b) \in R_1$ $(b, k) \in R_2$ |
| $(i, k) \in R_2 \circ R_1$ | $(i, k) \in R_2 \circ R_1$ |

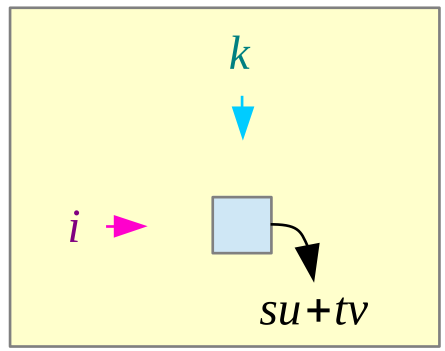
# Necessary Part

$$A_1 A_2 = \begin{bmatrix}1&0\\0&1\\1&1\end{bmatrix}\begin{bmatrix}1&1&0\\0&1&1\end{bmatrix} = \begin{bmatrix}1&1&0\\0&\boxed{1}&1\\1&2&1\end{bmatrix}$$

$$A_1 = \begin{array}{c|c} & \begin{array}{cc} a & b \end{array} \\ \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix}1&0\\0&1\\1&1\end{bmatrix}\end{array}$$

$$A_2 = \begin{array}{c|c} & \begin{array}{ccc} x & y & z \end{array} \\ \begin{array}{c}a\\b\end{array} & \begin{bmatrix}1&1&0\\0&1&1\end{bmatrix}\end{array}$$




$i \in \{1, 2, 3\}$

$k \in \{x, y, z\}$

$s \in \{0, 1\}$

$t \in \{0, 1\}$

$u \in \{0, 1\}$

$v \in \{0, 1\}$

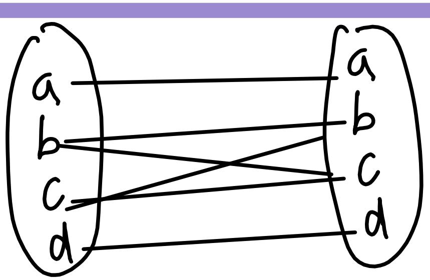
$$(i, k) \in R_2 \circ R_1 \Rightarrow$$

| | |
|---|---|
| $(i, a) \in R_1$ | $(i, b) \in R_1$ |
| $(a, k) \in R_2$ | $(b, k) \in R_2$ |
| $(s = 1)$ $(u = 1)$ | $(t = 1)$ $(v = 1)$ |
| $su = 1$ | $tv = 1$ |

$$su + tv \neq 0 \qquad (a_{ik} \neq 0)$$

# Transitivity Test Examples (1)

$R \in \{(a,a),\ (b,b),\ (c,c),\ (d,d),\ (b,c),\ (c,b)\}$




$R \circ R \in \{(a,a),\ (b,b),\ (c,c),\ (d,d),\ (b,c),\ (c,b)\}$

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$\|$ ? → transitive

$$A^2 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \color{red}{1} & \color{red}{1} & 0 \\ 0 & \color{red}{1} & \color{red}{1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$A\,A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \color{red}{2} & \color{red}{2} & 0 \\ 0 & \color{red}{2} & \color{red}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$A = A^2$

$A * A = A^2.$

set non-zero element to 1

$A^2 = A^3 = A$

# Transitivity Test Examples (2)

```
(%i8) A1:matrix(
        [1,0,0,0],
        [0,1,1,0],
        [0,1,1,0],
        [0,0,0,1]
      );
```

(%o8) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

```
(%i9) A2: A1.A1;
```

(%o9) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$A = A^2$

```
(%i10) A3: A2.A1;
```

(%o10) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

```
(%i11) A4: A3.A1;
```

(%o11) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

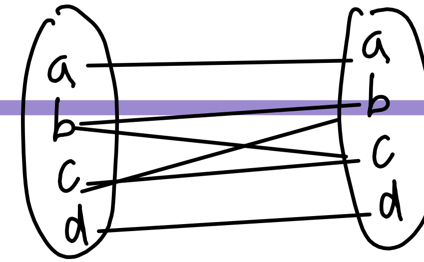
```
(%i12) A5: A4.A1;
```

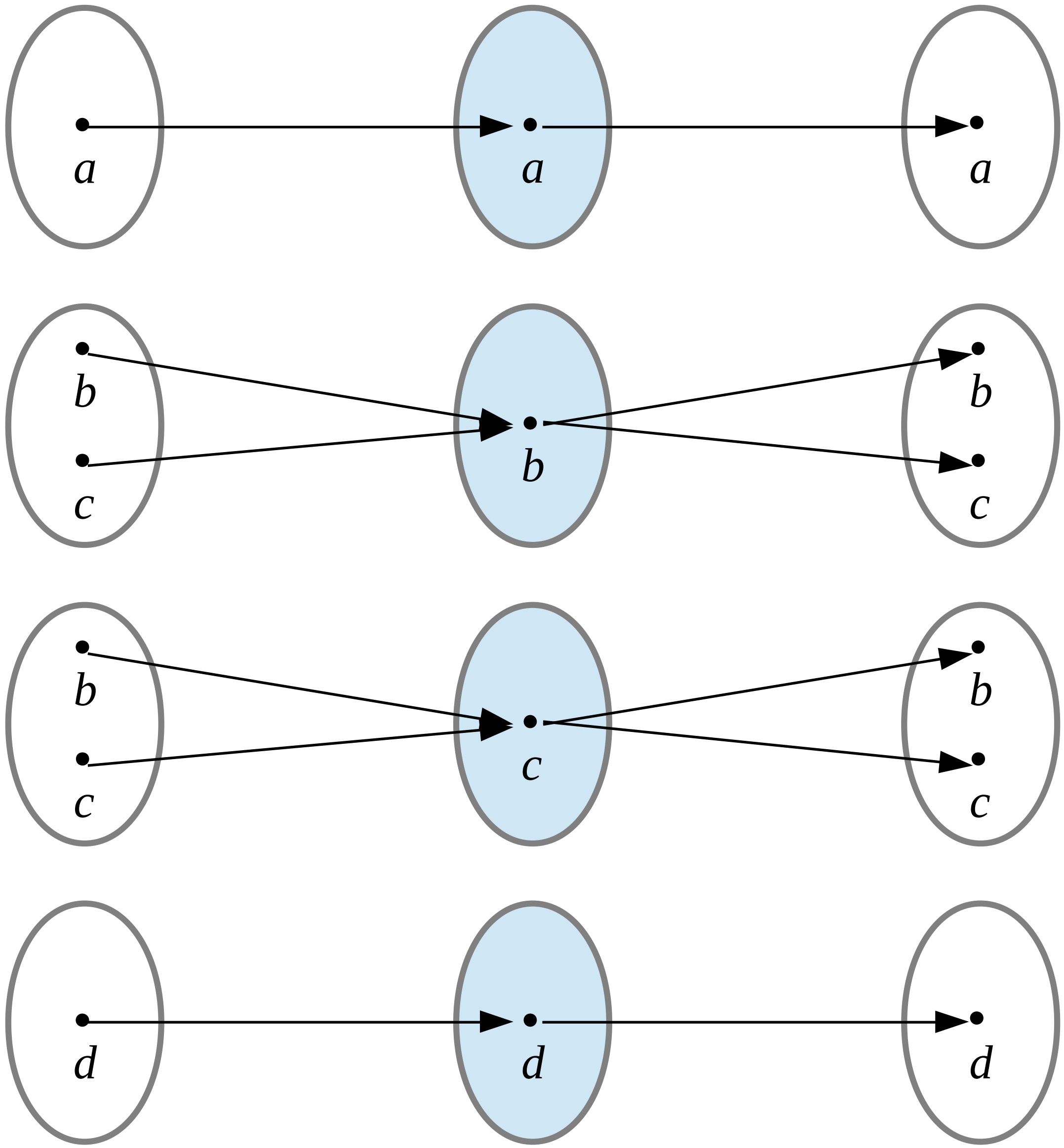
(%o12) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 16 & 16 & 0 \\ 0 & 16 & 16 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

```
(%i13) A6: A5.A1;
```

(%o13) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 32 & 32 & 0 \\ 0 & 32 & 32 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

# Transitivity Test Examples (3)




$(a, a)$

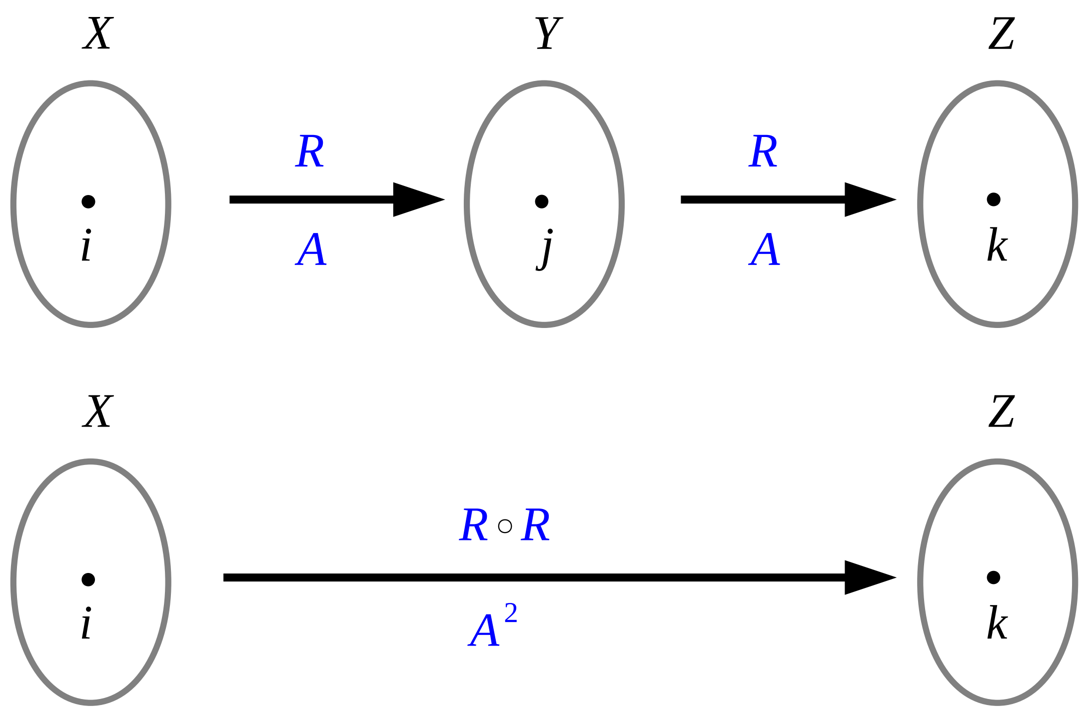
transitive

$(b, b)$
$(b, c)$
$(c, b)$
$(c, c)$

$(b, b)$
$(b, c)$
$(c, b)$
$(c, c)$

$(d, d)$

# Transitivity Test




$A = A^2 \rightarrow$

$A = A^3 \rightarrow$

$\cdots$

$A = A^n \rightarrow$

$\cdots$

$$\boxed{A = A^2} \quad \Longleftrightarrow \quad \boxed{\textit{transitive relation} \quad R}$$

# Transitivity Condition

$$A = \begin{array}{c|c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} a & b & c & d \\ e & f & \boxed{g} & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \end{array}$$

$g \neq 0$

$\neq 0 \quad \neq 0 \quad \neq 0 \quad \neq 0$

$e \cdot c + f \cdot g + g \cdot k + h \cdot o \neq 0$

?

$$A^2 = \begin{array}{c|c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} * & * & * & * \\ e & f & g & h \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \end{array} \begin{array}{c|c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} * & * & c & * \\ * & * & g & * \\ * & * & k & * \\ * & * & o & * \end{bmatrix} \end{array} = \begin{array}{c|c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \end{array}$$

$\text{nonzero } (i, j)^{th} \text{ element of } A^2 \Rightarrow \text{nonzero } (i, j)^{th} \text{ element of } A$

$A = A^2$

# A non-zero element of $A^2$

$(2,1)$ $\quad$ $(1,3) = (2,3)$

$$A^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ 1 & * & * & * & * \\ 2 & e & f & g & h \\ 3 & * & * & * & * \\ 4 & * & * & * & * \end{array} \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ 1 & * & * & c & * \\ 2 & * & * & g & * \\ 3 & * & * & k & * \\ 4 & * & * & o & * \end{array} = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ 1 & * & * & * & * \\ 2 & * & * & * & * \\ 3 & * & * & * & * \\ 4 & * & * & * & * \end{array} \Rightarrow A$$

| | | | | | |
|---|---|---|---|---|---|
| $a\,e = 1$ ∨ | $b\,f = 1$ ∨ | $c\,g = 1$ ∨ | $d\,h = 1$ | ⇒ | $a \cdot e + b \cdot f + c \cdot g + d \cdot h \neq 0$ |
| $(2,1) \in R$<br>$(1,3) \in R$ | $(2,2) \in R$<br>$(2,3) \in R$ | $(2,3) \in R$<br>$(3,3) \in R$ | $(2,4) \in R$<br>$(4,3) \in R$ | ⇒ | $(2,3) \in R$ |

$(2,1)\ (1,3)$ $\quad$ $(2,2)\ (2,3)$ $\quad$ $(2,3)(3,3)$ $\quad$ $(2,4)(4,3)$

$(2,3)$ $\quad$ $(2,3)$ $\quad$ $(2,3)$ $\quad$ $(2,3)$

$$\forall x, \forall y, \forall z\ \left[ \left( (x, y) \in R \ \wedge\ (y, z) \in R \right) \rightarrow (x, z) \in R \right]$$

# Binary Relations and Digraphs

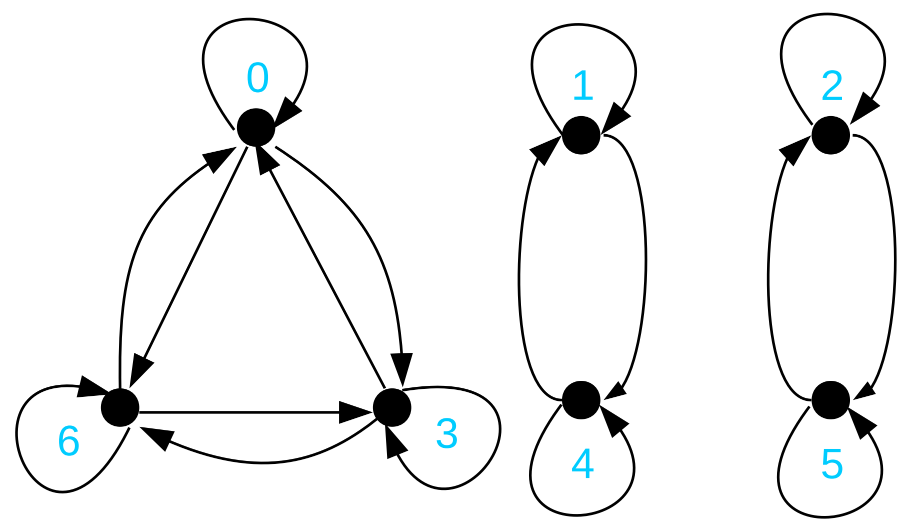
$$A = \{0,1,2,3,4,5,6\}$$

$$R \subset A \times A$$

$$R = \{\,(a,b) \mid a \equiv b \pmod 3\}$$

$$R = \begin{bmatrix} (0,0) & (0,1) & (0,2) & \cdots \\ (1,0) & (1,1) & (1,2) & \cdots \\ (2,0) & (2,1) & (2,2) & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix}$$

$$R = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$




http://www.math.fsu.edu/~pkirby/mad2104/SlideShow/s7_1.pdf

# Cartesian product

$A \times A = \{$
(0,0) (0,1) (0,2) (0,3) (0,4) (0,5) (0,6)
(1,0) (1,1) (1,2) (1,3) (1,4) (1,5) (1,6)
(2,0) (2,1) (2,2) (2,3) (2,4) (2,5) (2,6)
(3,0) (3,1) (3,2) (3,3) (3,4) (3,5) (3,6)
(4,0) (4,1) (4,2) (4,3) (4,4) (4,5) (4,6)
(5,0) (5,1) (5,2) (5,3) (5,4) (5,5) (5,6)
(6,0) (6,1) (6,2) (6,3) (6,4) (6,5) (6,6) $\}$

$$A = \{0,1,2,3,4,5,6\}$$

$$R \subset A \times A$$

$$R = \{(a,b) \mid a \equiv b \pmod 3\}$$

$R = \{$
(0,0) (0,3) (0,6)
(1,1) (1,4)
(2,2) (2,5)
(3,0) (3,3) (3,6)
(4,1) (4,4)
(5,2) (5,5)
(6,0) (6,3) (6,6) $\}$

$$A \times A = \left\{ \begin{array}{ccccccc} (0,0) & (0,1) & (0,2) & (0,3) & (0,4) & (0,5) & (0,6) \\ (1,0) & (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (1,6) \\ (2,0) & (2,1) & (2,2) & (2,3) & (2,4) & (2,5) & (2,6) \\ (3,0) & (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (3,6) \\ (4,0) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) & (4,6) \\ (5,0) & (5,1) & (5,2) & (5,3) & (5,4) & (5,5) & (5,6) \\ (6,0) & (6,1) & (6,2) & (6,3) & (6,4) & (6,5) & (6,6) \end{array} \right\}$$

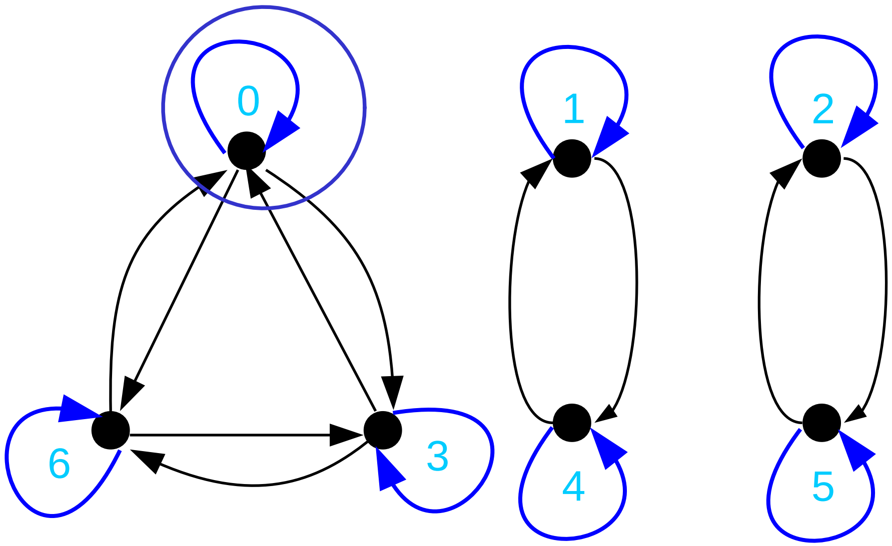
# Reflexive Relation

$$A = \{0,1,2,3,4,5,6\}$$

$$R \subset A \times A$$

$$R = \{(a,b) \mid a \equiv b \pmod 3\}$$

$$R = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



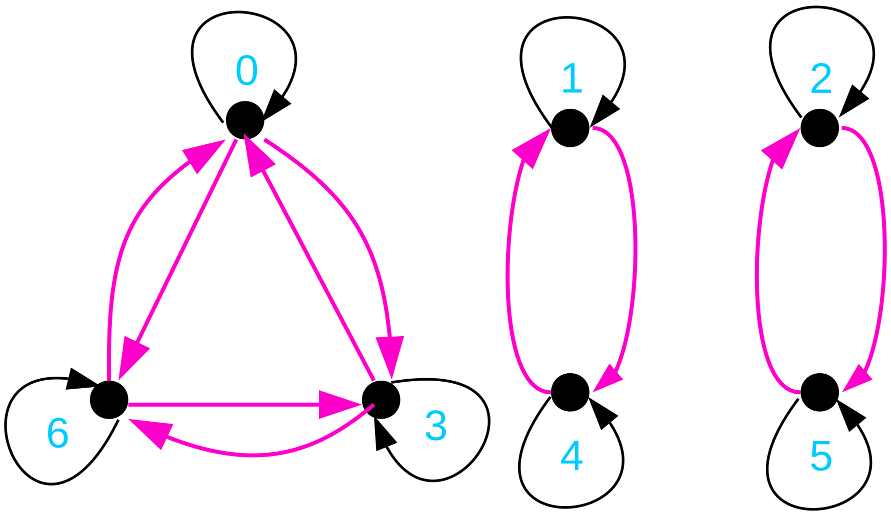
# Symmetric Relation

$$A = \{0, 1, 2, 3, 4, 5, 6\}$$

$$R \subset A \times A$$

$$R = \{(a, b) \mid a \equiv b \ (mod\ 3)\}$$

$$R = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



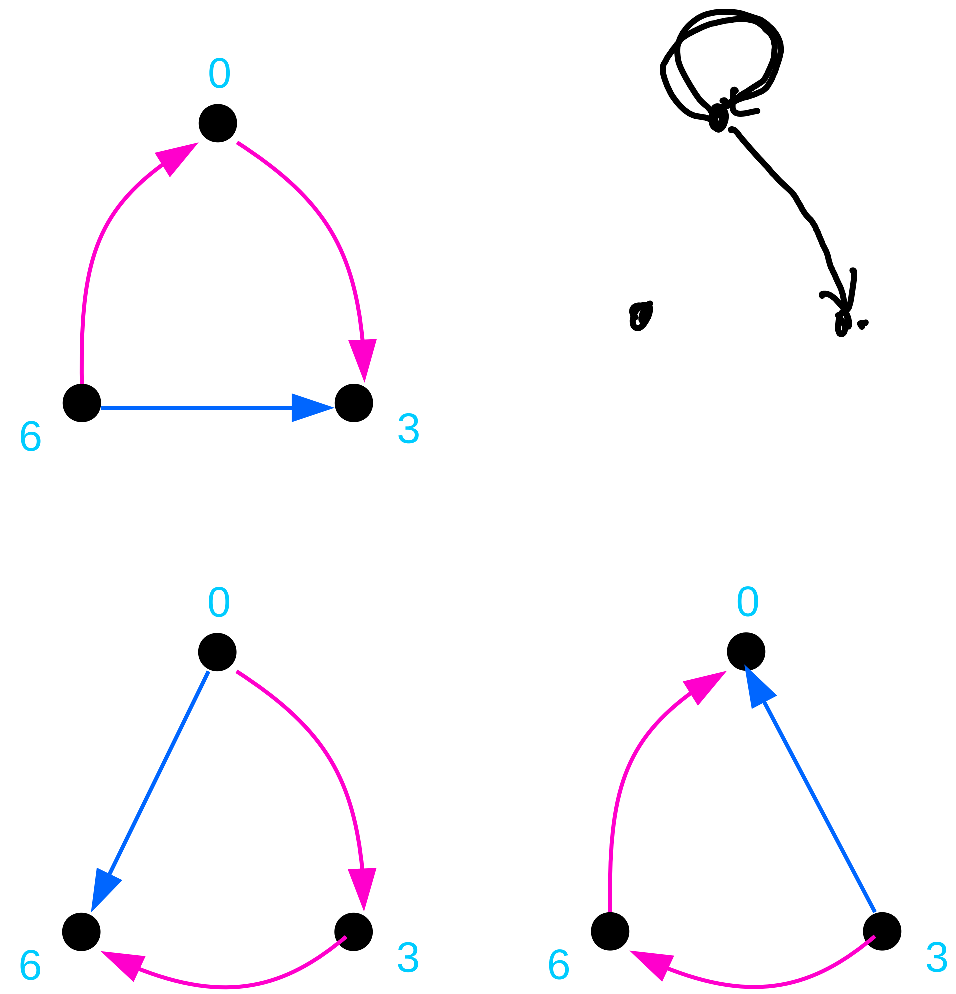
# Transitive Relation

$$A = \{0,1,2,3,4,5,6\}$$

$$R \subset A \times A$$

$$R = \{(a,b) \mid a \equiv b \pmod 3\}$$

$$RR = \begin{array}{c c}
 & \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} &
\begin{bmatrix}
3 & 0 & 0 & 3 & 0 & 0 & 3 \\
0 & 2 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 2 & 0 \\
3 & 0 & 0 & 3 & 0 & 0 & 3 \\
0 & 2 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 2 & 0 \\
3 & 0 & 0 & 3 & 0 & 0 & 3
\end{bmatrix}
\end{array}$$



# Transitive Relation

```
(%i2) R:matrix(
        [1,0,0,1,0,0,1],
        [0,1,0,0,1,0,0],
        [0,0,1,0,0,1,0],
        [1,0,0,1,0,0,1],
        [0,1,0,0,1,0,0],
        [0,0,1,0,0,1,0],
        [1,0,0,1,0,0,1]
      );
```

$$(\%o2)\quad \begin{bmatrix} 1&0&0&1&0&0&1\\ 0&1&0&0&1&0&0\\ 0&0&1&0&0&1&0\\ 1&0&0&1&0&0&1\\ 0&1&0&0&1&0&0\\ 0&0&1&0&0&1&0\\ 1&0&0&1&0&0&1 \end{bmatrix}$$

$A = A^2$

```
(%i4) R2: R.R;
```

$$(\%o4)\quad \begin{bmatrix} 3&0&0&3&0&0&3\\ 0&2&0&0&2&0&0\\ 0&0&2&0&0&2&0\\ 3&0&0&3&0&0&3\\ 0&2&0&0&2&0&0\\ 0&0&2&0&0&2&0\\ 3&0&0&3&0&0&3 \end{bmatrix}$$

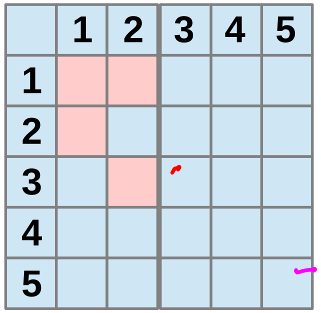
transitive

```
(%i7) R3: R.R.R;
```

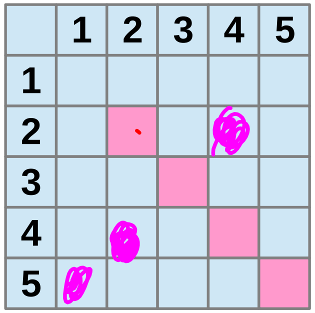
$$(\%o7)\quad \begin{bmatrix} 9&0&0&9&0&0&9\\ 0&4&0&0&4&0&0\\ 0&0&4&0&0&4&0\\ 9&0&0&9&0&0&9\\ 0&4&0&0&4&0&0\\ 0&0&4&0&0&4&0\\ 9&0&0&9&0&0&9 \end{bmatrix}$$

$R = R^2 \Rightarrow$ transitive

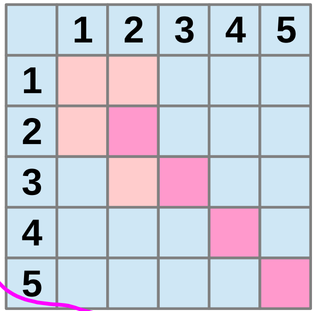
# Reflexive and Symmetric Closure



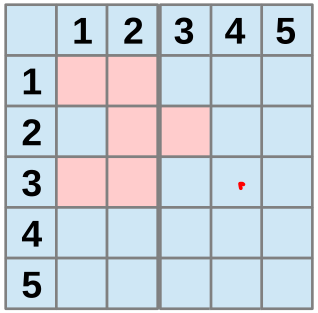
Not Reflexive R



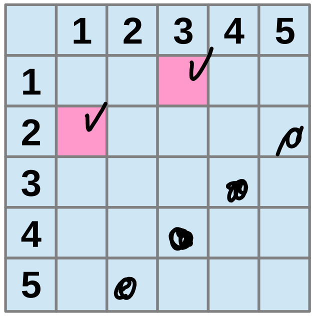
the minimal addition



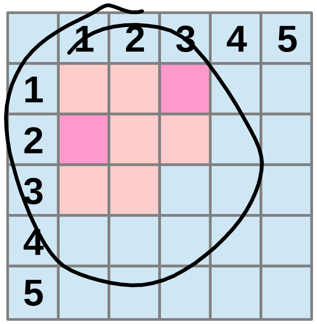
Reflexive Closure of R



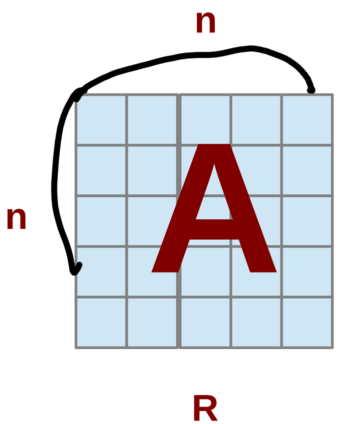
Not Symmetric R



the minimal addition



Symmetric Closure of R




$$R^* = \cup_{n=1}^{\infty} R^n$$

$$= R \cup R^2 \cup \cdots \cup R^n$$

$$A \vee A^2 \vee \cdots \vee A^n$$

set non-zero element to 1

# Transitive Closure Example 1

(%i9) A: matrix(
[1,0,1],
[0,1,0],
[1,1,0]
);

(%o9) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} = A$

not transitive

(%i11) A2: A.A;

(%o11) $\begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = A^2$

(%i12) A3: A.A.A;

(%o12) $\begin{bmatrix} 3 & 2 & 2 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix} = A^3$

$A^1 + A^2 + A^3$

(%i13) A+A2+A3;

(%o13) $\begin{bmatrix} 6 & 3 & 4 \\ 0 & 3 & 0 \\ 4 & 4 & 2 \end{bmatrix}$

(%i18) A4: A.A.A.A;

(%o18) $\begin{bmatrix} 5 & 4 & 3 \\ 0 & 1 & 0 \\ 3 & 3 & 2 \end{bmatrix}$

(%i20) A5: A.A.A.A.A;

(%o20) $\begin{bmatrix} 8 & 7 & 5 \\ 0 & 1 & 0 \\ 5 & 5 & 3 \end{bmatrix}$

(%i19) matrix(
[1,1,1],
[0,1,0],
[1,1,1]
);

not transitive

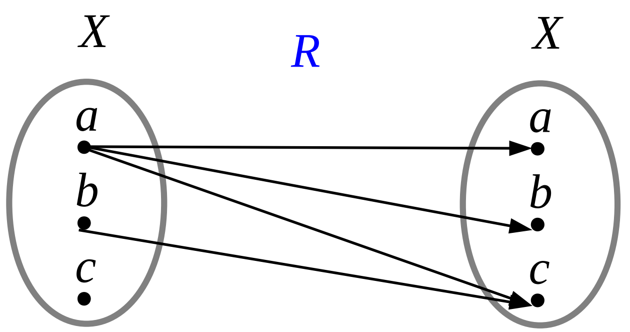
(%o19) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ transitive closure of A

transitive closure

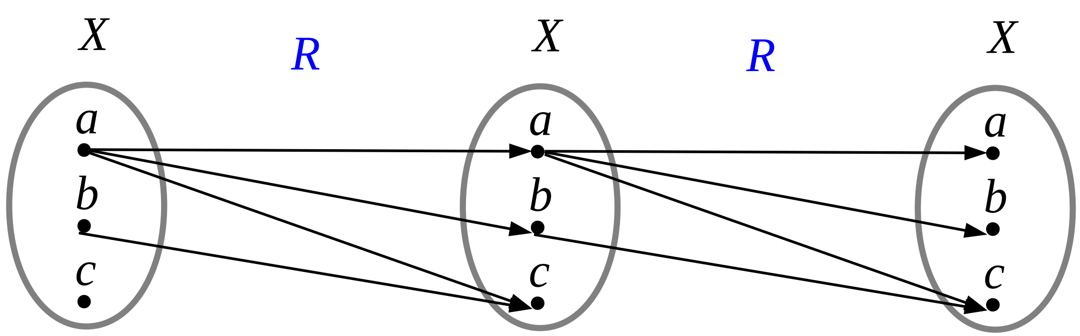
$A \neq A^2$

$A \neq tc(A)$ ⇨ not transitive

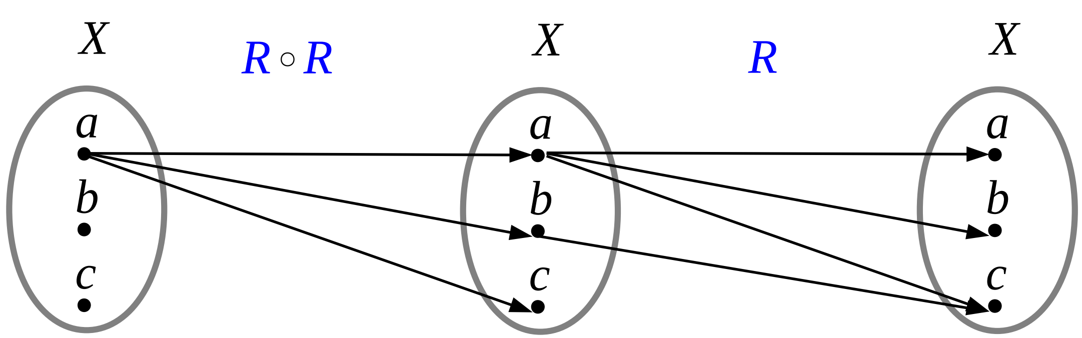
# Transitive Closure Example 2-a




$$A = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

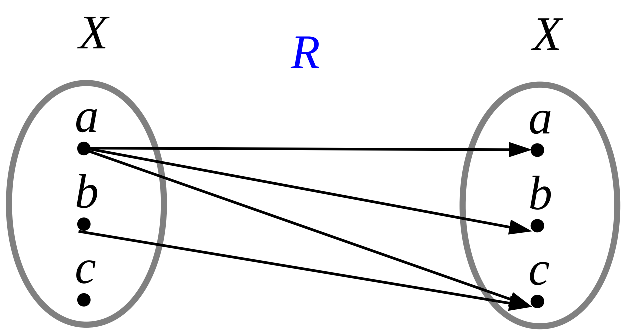


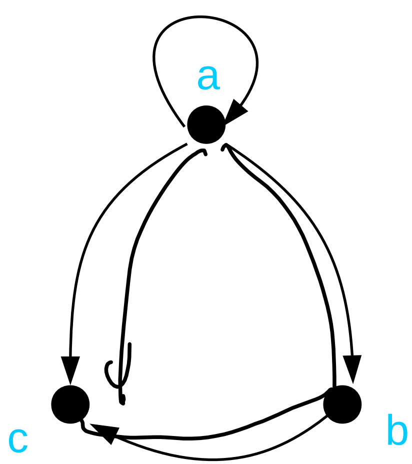

$$A^2 = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$




$$A^3 = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

# Transitive Closure Example 2-b




$$A = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$A^2 = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}.$$

$$A^3 = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$A^* = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$A^1 + A^2 + A^3 \Rightarrow$ t. closure

$$A \neq A^2$$

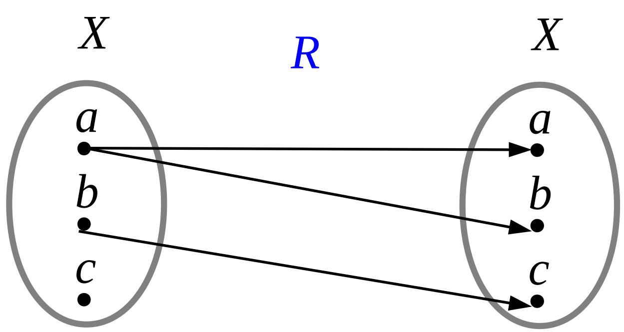
But the <u>transitive</u> <u>closure</u> of A is equal to A

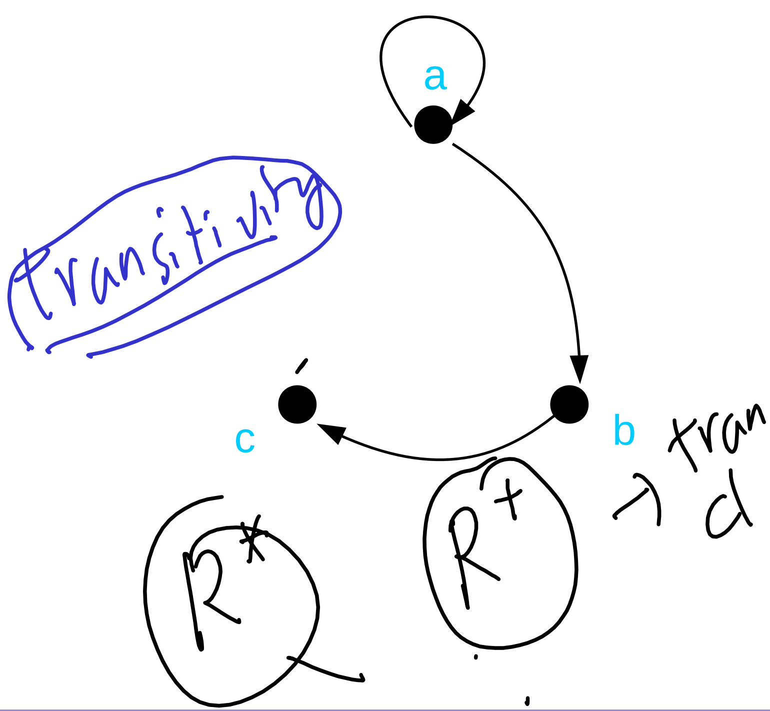
⇨ transitive

$$A \neq A^2$$

$$A = tc(A)$$ ⇨ transitive

# Transitive Closure Example 3




$$A = \begin{array}{c|ccc} & a & b & c \\ \hline a & 1 & 1 & 0 \\ b & 0 & 0 & 1 \\ c & 0 & 0 & 0 \end{array}$$

$$A^2 = \begin{array}{c|ccc} & a & b & c \\ \hline a & 1 & 1 & 1 \\ b & 0 & 0 & 0 \\ c & 0 & 0 & 0 \end{array}$$

$$A^3 = \begin{array}{c|ccc} & a & b & c \\ \hline a & 1 & 1 & 1 \\ b & 0 & 0 & 0 \\ c & 0 & 0 & 0 \end{array}$$

$$A^* = \begin{array}{c|ccc} & a & b & c \\ \hline a & 1 & 1 & 1 \\ b & 0 & 0 & 1 \\ c & 0 & 0 & 0 \end{array}$$

$A \neq A^2$

And the transitive closure of A is not equal to A

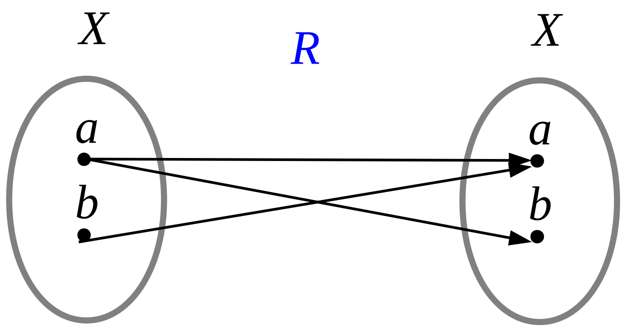
⇨ not transitive

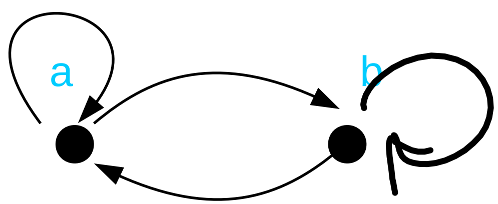
$A \neq A^2$

$A \neq tc(A)$

⇨ not transitive

# Transitive Closure Example 4




$$A = \begin{array}{c c} & \begin{array}{cc} a & b \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \end{array}$$

$$A^2 = \begin{array}{c c} & \begin{array}{cc} a & b \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \end{array}$$

$$A^* = \begin{array}{c c} & \begin{array}{cc} a & b \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \end{array}$$

$$A \neq A^2$$

$$A \neq tc(A)$$

⇨ not transitive

$$
\begin{tikzcd}
 & 1 & \\
2 \arrow[ur] \arrow[rr] & & 3 \arrow[ul]
\end{tikzcd}
$$

$$
A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}
$$

transitive

```
(%i10) B:matrix(
        [0,0,0],
        [1,0,1],
        [1,0,0]
       );
```

$$
(\%o10) \quad \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}
$$

```
(%i11) B2 : B. B;
```

$$
(\%o11) \quad \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
$$

```
(%i12) B3 : B. B. B;
```

$$
(\%o12) \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
$$

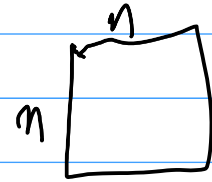
```
(%i13) B4 : B.B.B.B;
```

$$
(\%o13) \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
$$

$A \cdot A = A^2 A$ $\quad A^3 = A^2$

$A^4 = A^3$

$$\begin{cases} \boxed{A = A^2} & A: \underline{\text{transitive}} \\ A \neq A^2 \end{cases}$$




$$\begin{cases} A \vee A^2 \vee \cdots \vee A^n = A & A: \underline{\text{transitive}} \\ A \vee A^2 \vee \cdots \vee A^n \neq A & A: \boxed{\text{NOT transitive}} \end{cases}$$

$\Downarrow$

A의 transitive closure

**References**

[1] http://en.wikipedia.org/
[2]

# Equivalence Relations (4A)

① Equivalence Relation EQ

② Partial Order Relation

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

# Equivalence Relation

a binary relation that is at the same time

- a reflexive relation,
- a symmetric relation and
- a transitive relation.

The relation "**is equal to**" is a primary example of an equivalence relation.

Thus for any numbers a, b, and c:

**a=a** (reflexive property),
if **a=b** then **b=a** (symmetric property), and
if **a=b** and **b=c** then **a=c** (transitive property).

Any equivalence relation, as a consequence of the reflexive, symmetric, and transitive properties, provides a **partition** of a set into **equivalence classes**.

$\equiv \pmod 3$

$3 \equiv 3$

$3 \equiv 6 \quad 6 \equiv 3$

$3 \equiv 6 \quad 6 \equiv 9 \quad 3 \equiv 9$

https://en.wikipedia.org/wiki/Equivalence_relation

# Equivalence Relation Definition

A given binary relation ~ on a set **X**
is said to be an equivalence relation
if and only if it is reflexive, symmetric and transitive.

That is, for all a, b and c in **X**:

**a** ~ **a**. (Reflexivity)
**a** ~ **b** if and only if **b** ~ **a**. (Symmetry)
if **a** ~ **b** and **b** ~ **c** then **a** ~ **c**. (Transitivity)

**X** together with the relation ~ is called a **setoid**.
The **equivalence class** of a under ~, denoted **[a]**,
is defined as **[a] = { b ∈ X | a ~ b }**

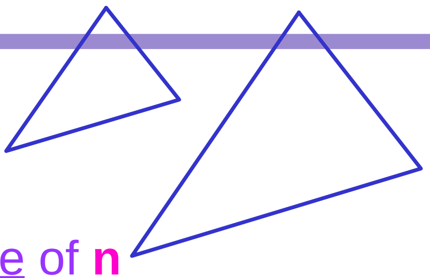
~

[a]

https://en.wikipedia.org/wiki/Equivalence_relation

# Congruent Modulo n

congruent



for a <u>positive</u> <u>integer</u> **n**, two numbers **a** and **b** are said to be **congruent modulo n**, if their difference **a − b** is an <u>integer</u> <u>multiple</u> of **n**
(that is, if there is an integer **k** such that **a − b = kn**).
This congruence relation is typically considered
when a and b are integers, and is denoted

**a ≡ b ( mod n )**

(some authors use = instead of ≡)

b % n

**a = b mod n** **// a = b % n**

(this generally means that "mod" denotes the modulo operation,
that is, that **0 ≤ a < n**).

The number n is called the **modulus** of the congruence.

https://en.wikipedia.org/wiki/Equivalence_relation

# Congruent Modulo n : Examples

$(38 - 14) = 24 = 2 \cdot 12$

For example,

38 ≡ 14 ( mod 12 )

$38 \not\equiv 14$

because 38 − 14 = 24, which is a multiple of 12,
or, equivalently, because both 38 and 14 have
the same remainder 2 when divided by 12.

$38 \equiv 14$

$14 \% 12 = 2$

The same rule holds for negative values:

−8 ≡ 7 ( mod 5 )
2 ≡ −3 ( mod 5 )
−3 ≡ −8 ( mod 5 )

$38 \% 12 = 2$

$36 \quad 38 - 3 \cdot 12 = 2$

https://en.wikipedia.org/wiki/Equivalence_relation

# Congruent Modulo n : Properties

The congruence relation satisfies all the conditions of an equivalence relation:

Reflexivity: **a ≡ a (mod n)**
Symmetry: **a ≡ b (mod n)** if and only if **b ≡ a (mod n)**
Transitivity: If **a ≡ b (mod n)** and **b ≡ c (mod n)**, then **a ≡ c (mod n)**

https://en.wikipedia.org/wiki/Equivalence_relation

# Equivalence Relation

| Equivalence Relation | $\Leftrightarrow$ | Reflexive Relation & Symmetric Relation & Transitive Relation |
|---|---|---|

$$A = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$$

$$R \subset A \times A$$

$$R = \{(a, b) \mid a \equiv b \pmod 3\}$$

# Equivalence Class

$$A = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$$

$$R \subset A \times A$$

$$R = \{ (a, b) \mid a \equiv b \ (mod\ 3) \}$$

$(0,0), (0,3), (0,6),$
$(3,0), (3,3), (3,6),$
$(6,0), (6,3), (6,6)$

$0 \sim 0,\ 0 \sim 3,\ 0 \sim 6,$
$3 \sim 0,\ 3 \sim 3,\ 3 \sim 6,$
$6 \sim 0,\ 6 \sim 3,\ 6 \sim 6$

$[0] = \{0, 3, 6\}$

$[0] \subset A$

$(1,1), (1,4), (1,7),$
$(4,1), (4,4), (4,7),$
$(7,1), (7,4), (7,7)$

$1 \sim 1,\ 1 \sim 4,\ 1 \sim 7,$
$4 \sim 1,\ 4 \sim 4,\ 4 \sim 7,$
$7 \sim 1,\ 7 \sim 4,\ 7 \sim 7$

$[1] = \{1, 4, 7\}$

$[1] \subset A$

$(2,2), (2,5), (2,8),$
$(5,2), (5,5), (5,8),$
$(8,2), (8,5), (8,8)$

$2 \sim 2,\ 2 \sim 5,\ 2 \sim 8,$
$5 \sim 2,\ 5 \sim 5,\ 5 \sim 8,$
$8 \sim 2,\ 8 \sim 5,\ 8 \sim 8$

$[2] = \{2, 5, 8\}$

$[2] \subset A$

# Partitions

$$A = \{0,1,2,3,4,5,6,7,8\}$$

$$R \subset A \times A$$

$$R = \{(a,b) \mid a \equiv b \pmod 3\}$$

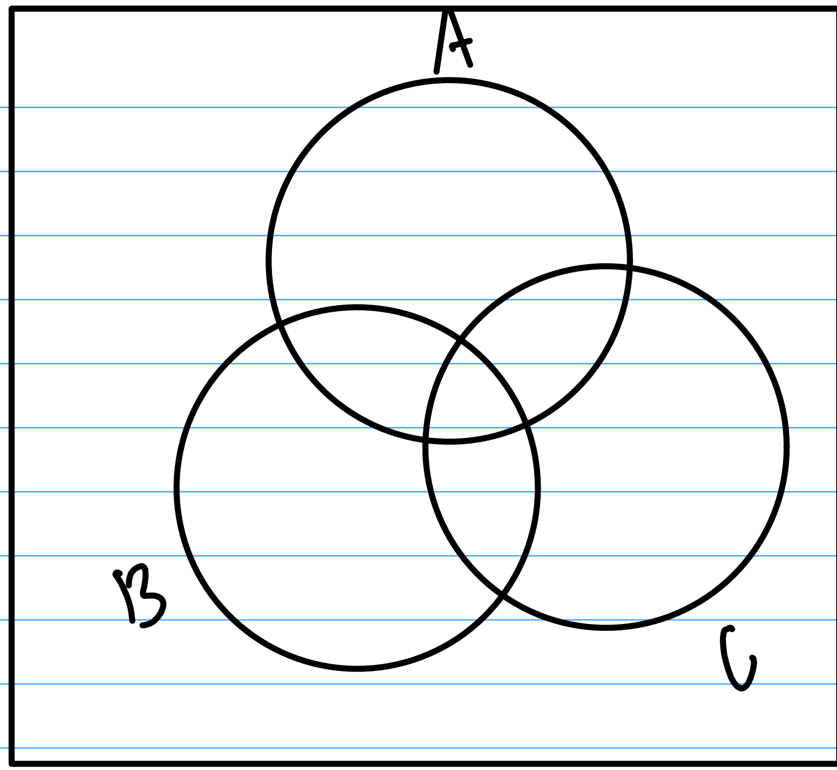
$$[0] = \{0, 3, 6\} \qquad [1] = \{1, 4, 7\} \qquad [2] = \{2, 5, 8\}$$

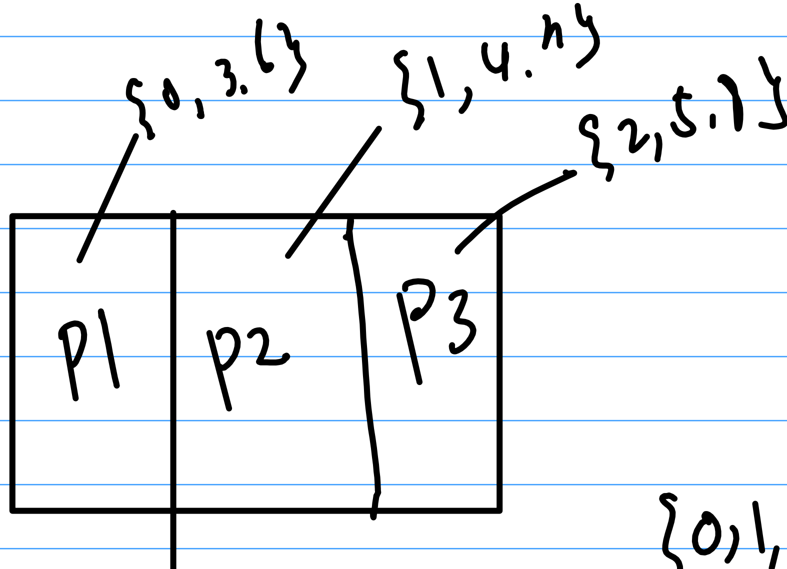
$$[0] \subset A \qquad [1] \subset A \qquad [2] \subset A$$

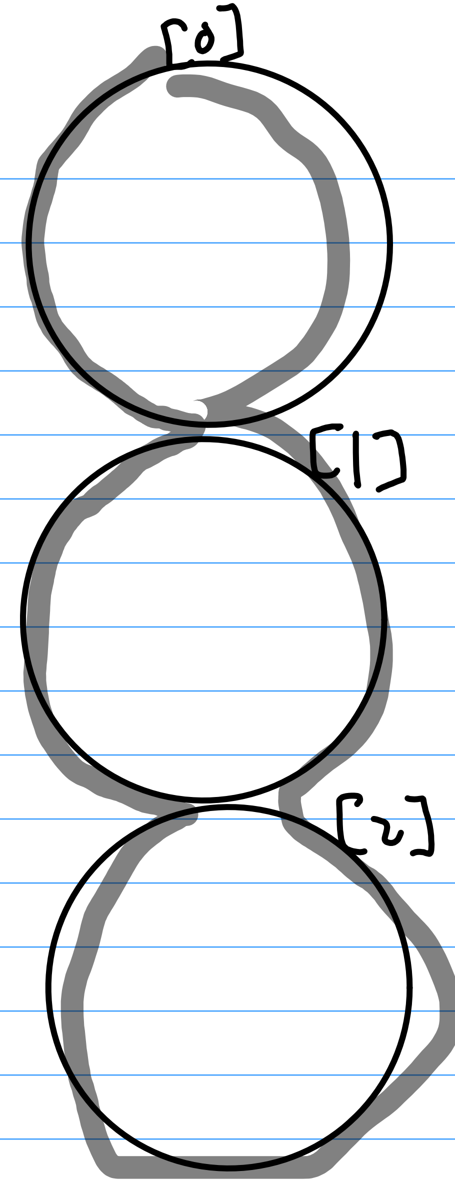
$$[0] \cap [1] = \varnothing \qquad [1] \cap [2] = \varnothing \qquad [2] \cap [0] = \varnothing$$

$$[0] \cup [1] \cup [2] = \{0, 3, 6\} \cup \{1, 4, 7\} \cup \{2, 5, 8\} = \{0,1,2,3,4,5,6,7,8\} = A$$

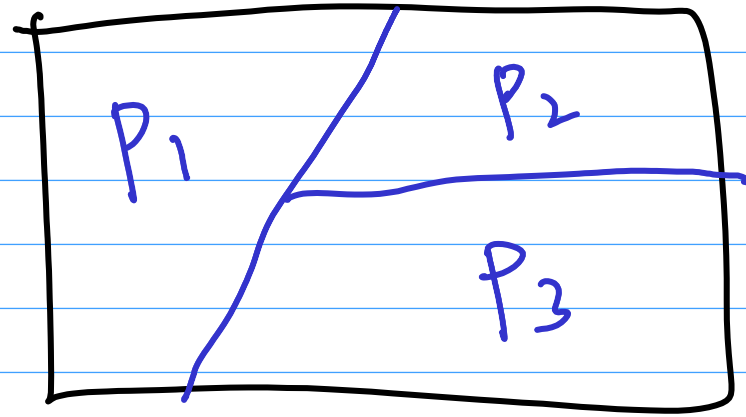
$$[0] = [3] = [6] \qquad [1] = [4] = [7] \qquad [2] = [5] = [8]$$



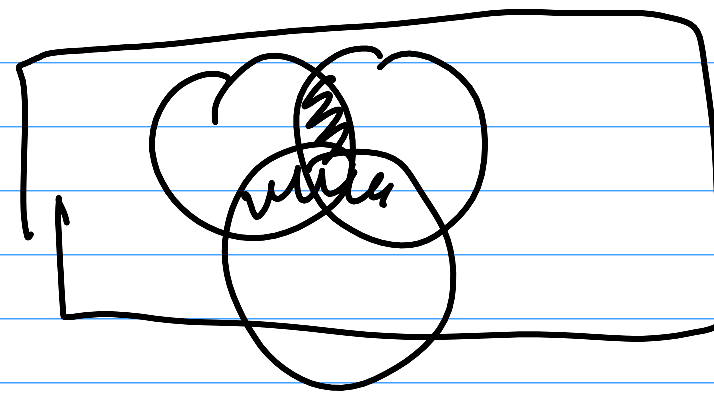




$\{0, 1, 2, \ldots, 8\} = A$




partition



# Equivalence Relation Examples

$$A = \{0,1,2,3,4,5,6,7,8\}$$

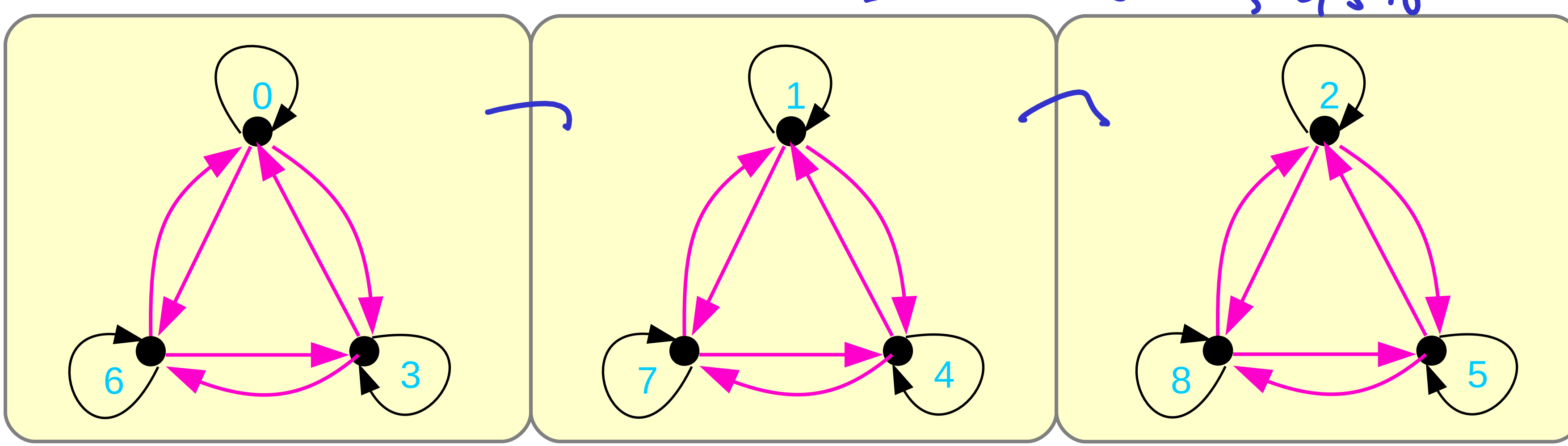
$$R \subset A \times A$$

$$R = \{(a,b) \mid a \equiv b \pmod 3\}$$

$$RR = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

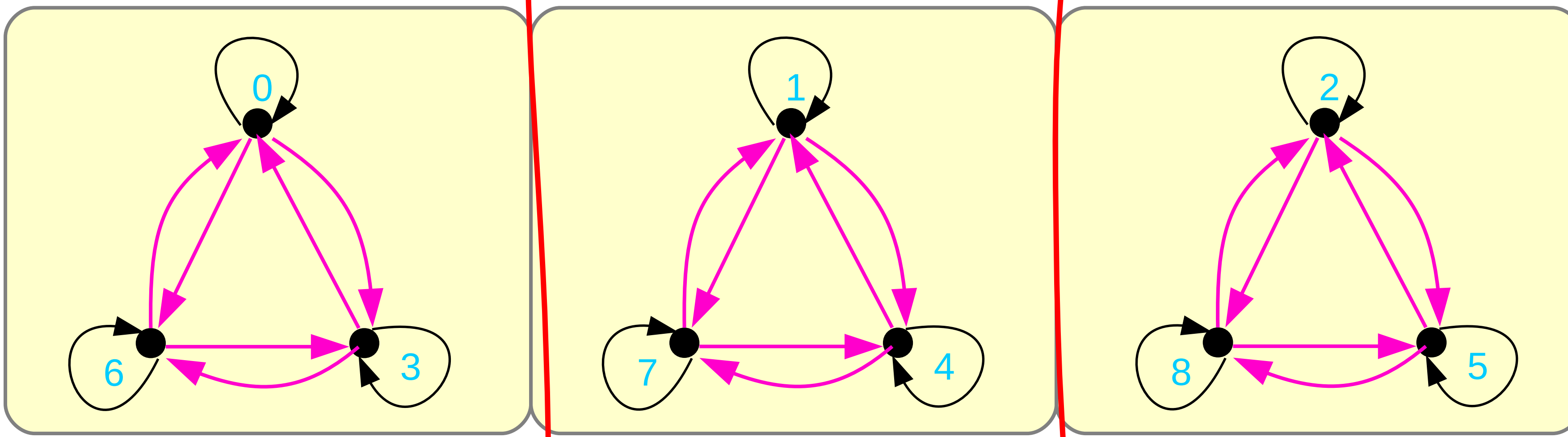
partition

$[0] = \{0, 3, 6\}$ $[1] = \{1, 4, 7\}$ $[2] = \{2, 5, 8\}$



# Equivalence Classes

[0] Class #1 | [1] Class #2 | [2] Class #3




$$\begin{array}{c|c} & \begin{array}{ccc} 0 & 3 & 6 \end{array} \\ \begin{array}{c} 0 \\ 3 \\ 6 \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array} \qquad \begin{array}{c|c} & \begin{array}{ccc} 1 & 4 & 7 \end{array} \\ \begin{array}{c} 1 \\ 4 \\ 7 \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array} \qquad \begin{array}{c|c} & \begin{array}{ccc} 2 & 5 & 8 \end{array} \\ \begin{array}{c} 2 \\ 5 \\ 8 \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array}$$

partition $P_1 = \{0, 3, 6\}$ partition $P_2 = \{1, 4, 7\}$ partition $P_3 = \{2, 5, 8\}$


Young Won Lim
4/12/18

# Equivalence Class

$$A = \mathbf{Z}^{+} = \{0, 1, 2, 3, 4, 5, 6, \cdots\}$$

$$R \subset A \times A$$

$$R = \{(a, b) \mid a \equiv b \ (mod\ 3)\}$$

| | | |
|---|---|---|
| $\{0,\ 3,\ 6,\ 9,\ \cdots\}$ | $[0]$ | $[33]$ |
| $\{1,\ 4,\ 7,\ 10,\ \cdots\}$ | $[1]$ | $[331]$ |
| $\{2,\ 5,\ 8,\ 11,\ \cdots\}$ | $[2]$ | $[3332]$ |

https://www.cse.iitb.ac.in/~nutan/courses/cs207-12/notes/lec7.pdf

## References


[1] http://en.wikipedia.org/
[2]

# Partial Oder Relations (5A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

# ~~Equivalence~~ Relation

$a \succsim b$

Partial Order

A (non-strict) **partial order** is a binary relation **≤** over a set P satisfying particular axioms.
When **a ≤ b**, we say that **a** is related to **b**.
(This does not imply that **b** is also related to **a**,
because the relation need not be symmetric.)

$3 \le 4$ (T) $\quad 4 \le 3$ (F) $\qquad a \precsim b, \quad b \precsim a \qquad \succ \qquad a \succsim b$

That is, for all **a**, **b**, and **c** in P, it must satisfy:

$3 \le 3 \quad 3 \le 3 \qquad 3 \le 3$

**a ≤ a** (reflexivity)
if **a ≤ b** and **b ≤ a**, then $\boxed{\textbf{a = b}}$ (antisymmetry)
if **a ≤ b** and **b ≤ c**, then **a ≤ c** (transitivity)

$3 \le 4 \qquad 4 \le 5 \qquad 3 \le 5 \qquad 3 \le 4 \qquad \cancel{4 \le 3}$

$3 \le 4 \qquad 4 \le 5 \qquad 3 \le 5$

https://en.wikipedia.org/wiki/Hasse_diagram

# Equivalence Relation

The axioms for a non-strict partial order state that the relation **≤** is

**reflexive**: every element is related to itself.

**antisymmetric**: two distinct elements cannot be related in both directions

**transitive**: if a first element is related to a second element,
and, in turn, that element is related to a third element,
then the first element is related to the third element

https://en.wikipedia.org/wiki/Hasse_diagram

# Relation Examples (1)

PO

$x \geq y$

p.O set

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **1** | (1,1) | | | | |
| **2** | (2,1) | (2,2) | | | |
| **3** | (3,1) | (3,2) | (3,3) | | |
| **4** | (4,1) | (4,2) | (4,3) | (4,4) | |
| **5** | (5,1) | (5,2) | (5,3) | (5,4) | (5,5) |

reflexive
anti-symmetric
transitive

$x > y$

~~partial ord~~

p.O set

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **1** | | | | | |
| **2** | (2,1) | | | | |
| **3** | (3,1) | (3,2) | | | |
| **4** | (4,1) | (4,2) | (4,3) | | |
| **5** | (5,1) | (5,2) | (5,3) | (5,4) | |

~~reflexive~~
transitive
anti-symmetric

https://en.wikipedia.org/wiki/Cartesian_product

~~PO~~

transitive

(%i2)
```
A:matrix(
  [0,0,0,0,0],
  [1,0,0,0,0],
  [1,1,0,0,0],
  [1,1,1,0,0],
  [1,1,1,1,0]
);
```

(%o2) $A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$

(%i3)
```
A2 : A.A;
```

(%o3) $A^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 \end{bmatrix}$

(%i6)
```
A3 : A.A.A;
```

(%o6) $A^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \end{bmatrix}$

$0 \to 1$

transitive ✗

$1 \to 0$

(%i7)
```
A4: A.A.A.A;
```

(%o7) $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} = A^4$

(%i8)
```
A5 : A.A.A.A.A;
```

(%o8) $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

Partial Order R

≡

- Reflexive
- anti-symmetric
- transitive

# Equivalence Relation

Partial order

| Equivalence Relation | ⟺ | Reflexive Relation & Anti-Symmetric Relation & Transitive Relation |
|---|---|---|


Young Won Lim

## References

[1] http://en.wikipedia.org/
[2]

# Algorithms – Bubble Sort (1B)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

# Bubble Sort Algorithm

**procedure** bubblesort($a_1$, … , $a_n$ : real numbers with $n \geq 2$)

**for** $i$ := 1 **to** $n$-1

  **for** $j$ := 1 **to** $n - i$

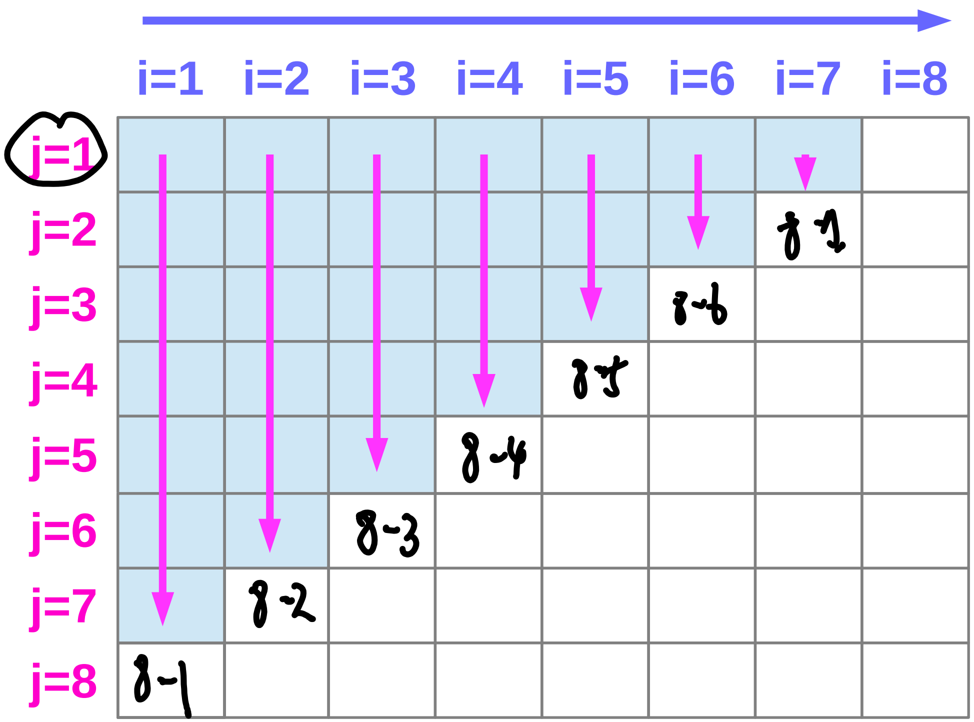
    **if** $a_j > a_{j+1}$ **then** interchange $a_j$ and $a_{j+1}$

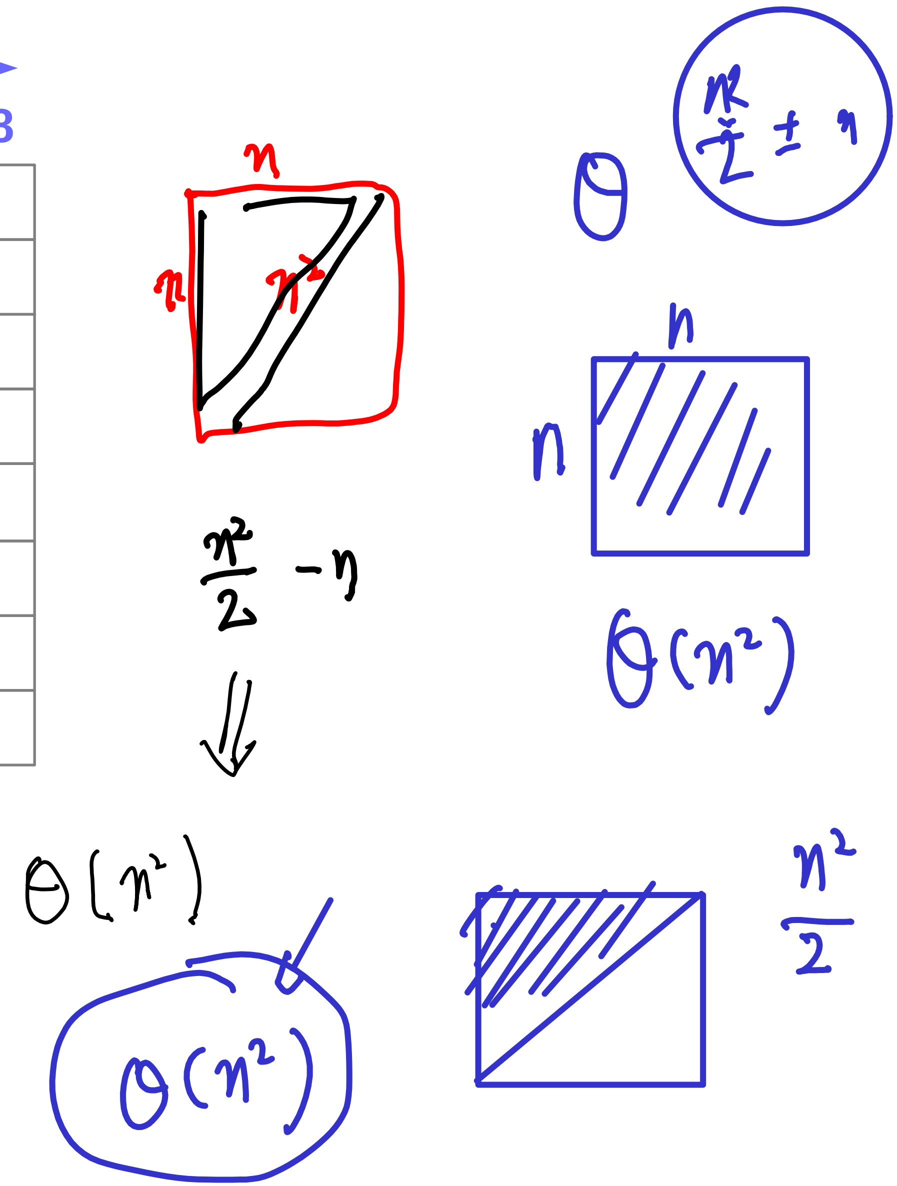
{$a_1$, …, $a_n$ is in increasing order}



Young Won Lim
4/5/18

# Nested loop iterations




$n=8$

**for** i := 1 **to** **n**-1 $8-1=7$

**for** j := 1 **to** **n** – i

# Input and Ouput

worst case

| | | | |
|---|---|---|---|
| $a_1$ | 44 | 88 | 11 |
| $a_2$ | 55 | 77 | 22 |
| $a_3$ | 22 | 66 | 33 |
| $a_4$ | 88 | 55 | 44 |
| $a_5$ | 66 | 44 | 55 |
| $a_6$ | 11 | 33 | 66 |
| $a_7$ | 77 | 22 | 77 |
| $a_8$ | 33 | 11 | 88 |

| | |
|---|---|
| $a_1$ | 11 |
| $a_2$ | 22 |
| $a_3$ | 33 |
| $a_4$ | 44 |
| $a_5$ | 55 |
| $a_6$ | 66 |
| $a_7$ | 77 |
| $a_8$ | 88 |

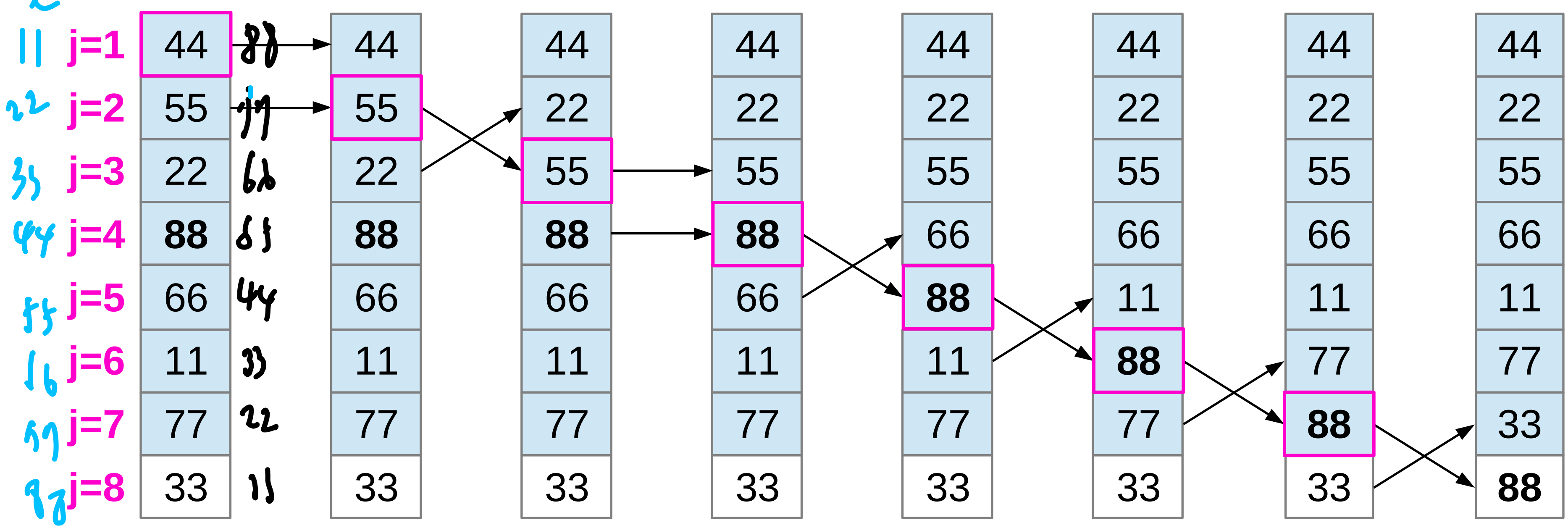
$a_1, \ldots, a_n$ : real numbers
with $n \geq 2$

$\{a_1, \ldots, a_n$ is in increasing order$\}$

# Step i=1

best $\Theta(n)$

worst $\Theta(n^2)$

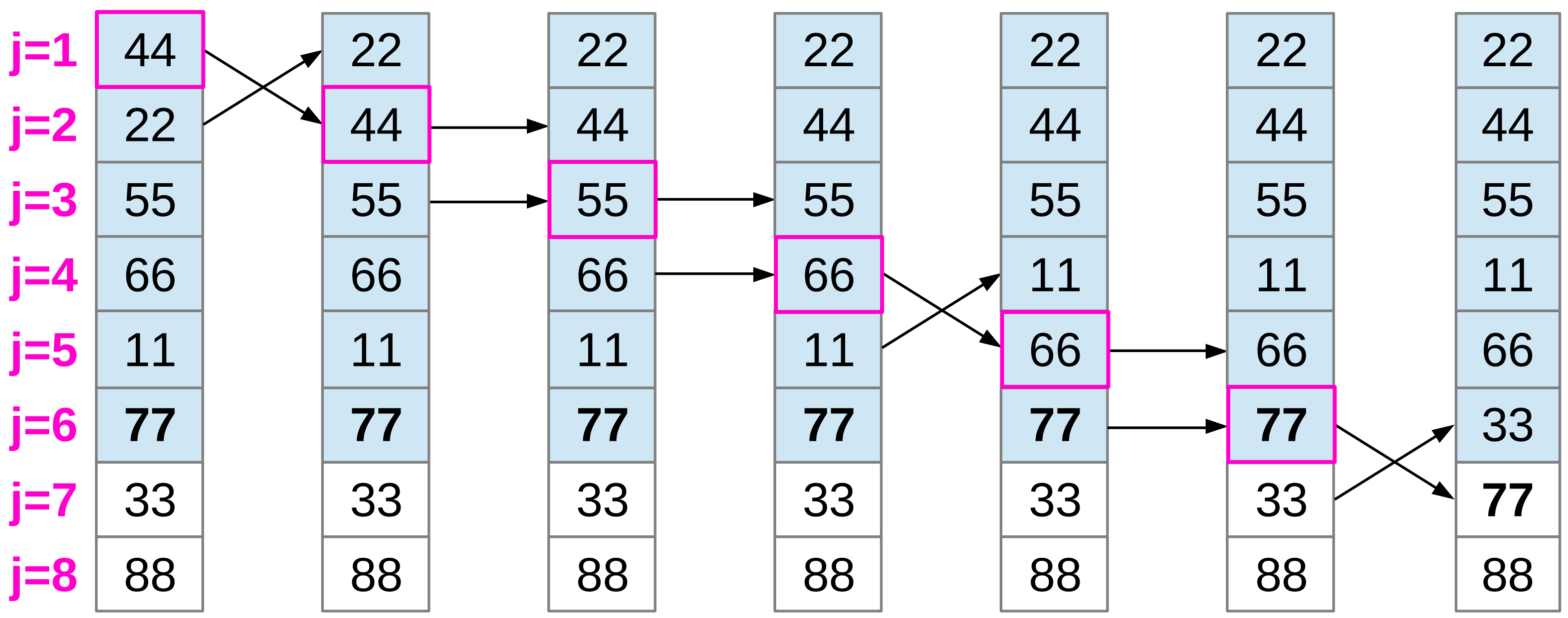



**for** $i$ := 1 **to** $n$-1

    **for** $j$ := 1 **to** $n - i$

        **if** $a_j > a_{j+1}$ **then** interchange $a_j$ and $a_{j+1}$

# Step i=2

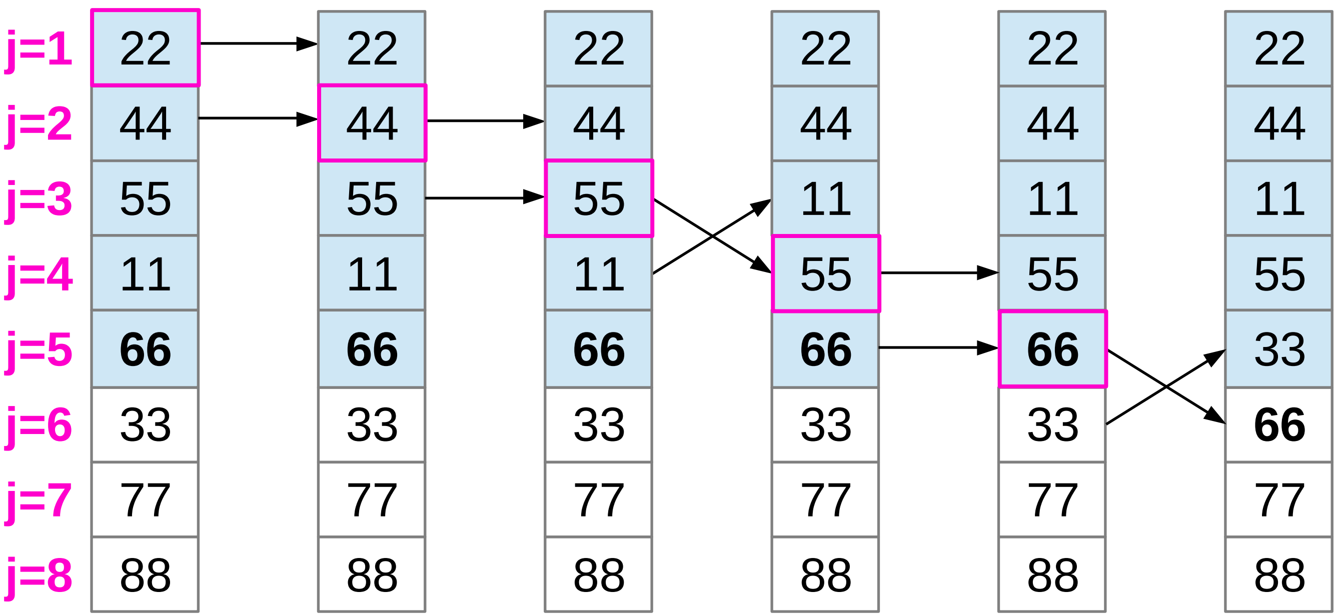



**for** i := 1 **to** n-1

  **for** j := 1 **to** n – i

    **if** $a_j > a_{j+1}$ **then** interchange $a_j$ and $a_{j+1}$

# Step i=3

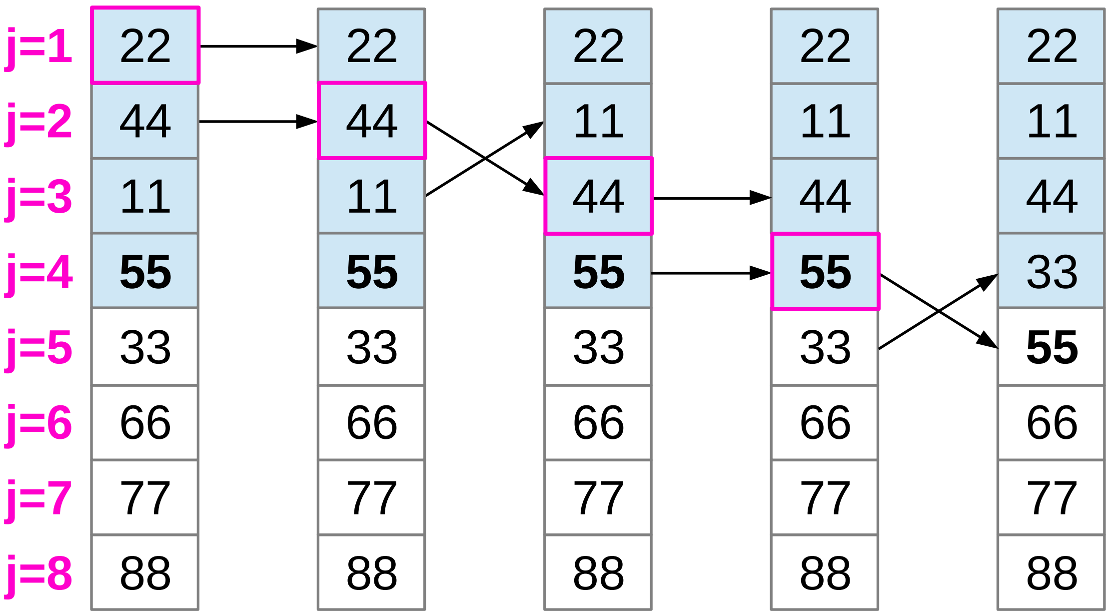



**for** i := 1 **to** **n**-1

  **for** j := 1 **to** **n** – i

    **if** $a_j > a_{j+1}$ **then** interchange $a_j$ and $a_{j+1}$

# Step i=4

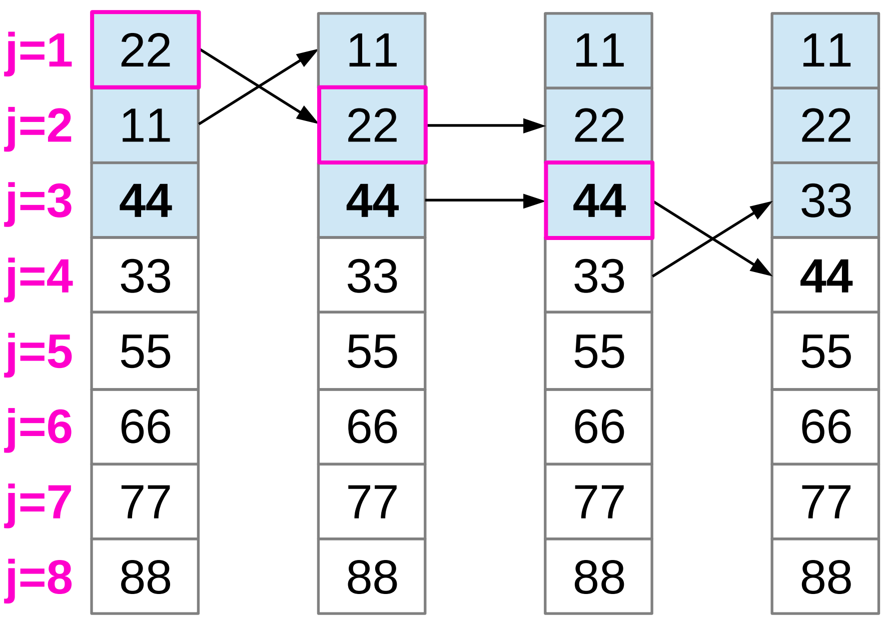



**for** $i$ := 1 **to** $n$-1

  **for** $j$ := 1 **to** $n - i$

    **if** $a_j > a_{j+1}$ **then** interchange $a_j$ and $a_{j+1}$

# Step i=5

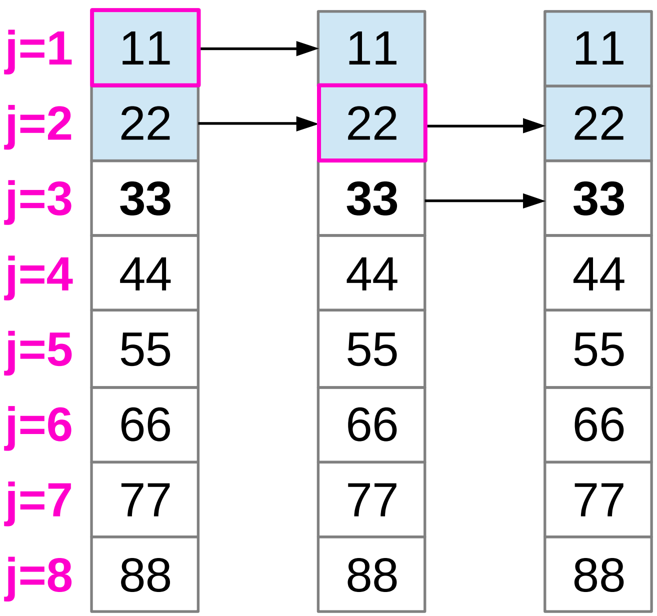



**for** i := 1 **to** **n**-1

  **for** j := 1 **to** **n** – i

    **if** $a_j > a_{j+1}$ **then** interchange $a_j$ and $a_{j+1}$

# Step i=6

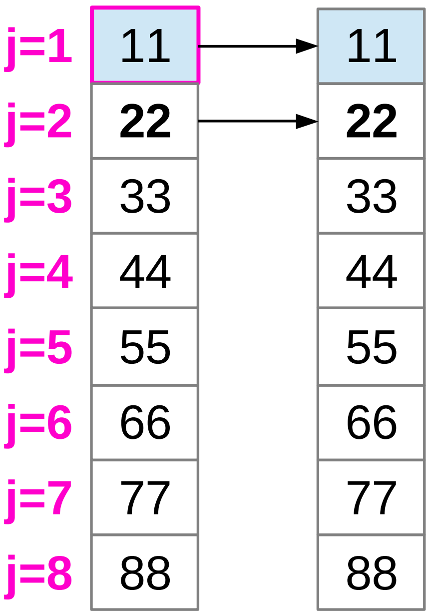



**for** $i$ := 1 **to** $n$-1

  **for** $j$ := 1 **to** $n - i$

    **if** $a_j > a_{j+1}$ **then** interchange $a_j$ and $a_{j+1}$

# Step i=7




**for** i := 1 **to** **n**-1

  **for** j := 1 **to** **n** – i

    **if** $a_j > a_{j+1}$ **then** interchange $a_j$ and $a_{j+1}$

# Summary

| | | i=1 | i=2 | i=3 | i=4 | i=5 | i=6 | i=7 |
|---|---|---|---|---|---|---|---|---|
| j=1 | 44 | 44 | 22 | 22 | 22 | 11 | 11 | 11 |
| j=2 | 55 | 22 | 44 | 44 | 11 | 22 | 22 | **22** |
| j=3 | 22 | 55 | 55 | 11 | 44 | 33 | **33** | 33 |
| j=4 | 88 | 66 | 11 | 55 | 33 | **44** | 44 | 44 |
| j=5 | 66 | 11 | 66 | 33 | **55** | 55 | 55 | 55 |
| j=6 | 11 | 77 | 33 | **66** | 66 | 66 | 66 | 66 |
| j=7 | 77 | 33 | **77** | 77 | 77 | 77 | 77 | 77 |
| j=8 | 33 | **88** | 88 | 88 | 88 | 88 | 88 | 88 |

**for** i := 1 **to** n-1

  **for** j := 1 **to** n – i

    **if** $a_j > a_{j+1}$ **then** interchange $a_j$ and $a_{j+1}$

## References


[1] http://en.wikipedia.org/
[2]

# Algorithms – Insertion Sort (1C)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

# Insertion Sort Algorithm

**procedure** insertion sort($a_1, \ldots, a_n$ : real numbers with $n \geq 2$)

**for** $j$ := 2 **to** $n$
    $i$ := 1
    **while** $a_j > a_i$
        $i$ := $i$ + 1
    $m$ := $a_j$
    **for** $k$ := 0 **to** $j - i - 1$
        $a_{j-k} = a_{j-k-1}$
    $a_i$ := $m$

{$a_1, \ldots, a_n$ is in increasing order}

7/0

$k$ = 0 0
0 1
0 2
0 3
0 ...

0
0, 1
0, 1, 2
0, 1, 2, 3

# Nested loop k – constraints

**for** $k$ := 0 **to** $j - i - 1$
$a_{j-k} = a_{j-k-1}$

$\boxed{j - i - 1 \geq 0}$ $\quad j \geq i + 1 \quad i \leq j - 1 \quad i < j$

$$a_{j-k} = a_{j-k-1}$$

$i < j$

| | | | |
|---|---|---|---|
| $(k=0)$ | $a_{j-0} = a_{j-0-1}$ | ⇒ | $a_j = a_{j-1}$ |
| $(k=1)$ | $a_{j-1} = a_{j-1-1}$ | ⇒ | $a_{j-1} = a_{j-2}$ |
| $(k=2)$ | $a_{j-2} = a_{j-2-1}$ | ⇒ | $a_{j-2} = a_{j-3}$ |
| | $\vdots = \vdots$ | | $\vdots = \vdots$ |
| $(k=j-i-1)$ | $a_{\cancel{j}-(\cancel{j}-i-1)} = a_{\cancel{j}-(\cancel{j}-i-\cancel{1})-\cancel{1}}$ | ⇒ | $a_{i+1} = a_i$ |

increasing index

# Nested loop k – rearranging for understanding

**for** $k$ := 0 **to** $j - i - 1$
  $a_{j-k} = a_{j-k-1}$

$j - i - 1 \geq 0$  $j \geq i + 1$  $\underline{i \leq j - 1}$  $\underline{i < j}$

$k=0, \; k=1, \; k=2,$

$$a_{j-k} = a_{j-k-1}$$

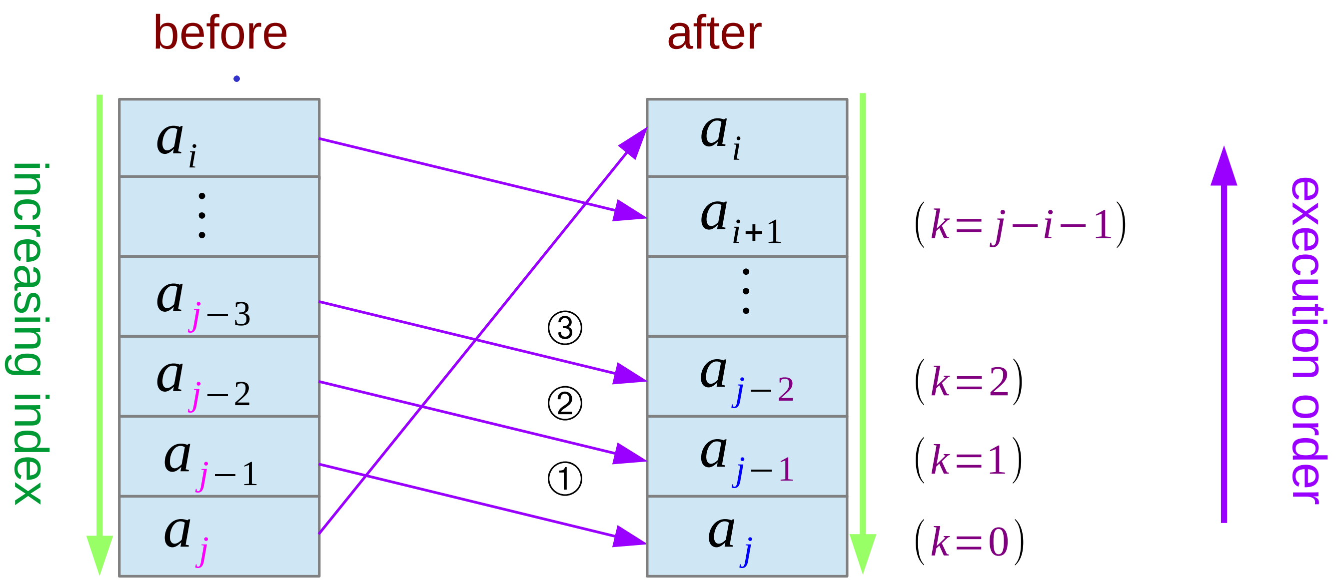
| | increasing index ↑ |
|---|---|
| $a_j = a_{j-1}$ | |
| $a_{j-1} = a_{j-2}$ | |
| $a_{j-2} = a_{j-3}$ | |
| $\vdots = \vdots$ | |
| $a_{i+1} = a_i$ | |

| increasing index ↓ | | execution order ↑ |
|---|---|---|
| $a_{i+1} = a_i$ | $(k = j - i - 1)$ | |
| $\vdots = \vdots$ | | |
| $a_{j-2} = a_{j-3}$ | $(k=2)$ | |
| $a_{j-1} = a_{j-2}$ | $(k=1)$ | |
| $a_j = a_{j-1}$ | $(k=0)$ | |

# Nested loop k – data movement

```
m := a_j
for k := 0 to j – i – 1
        a_{j-k} = a_{j-k-1}
a_i := m
```

$i < j$



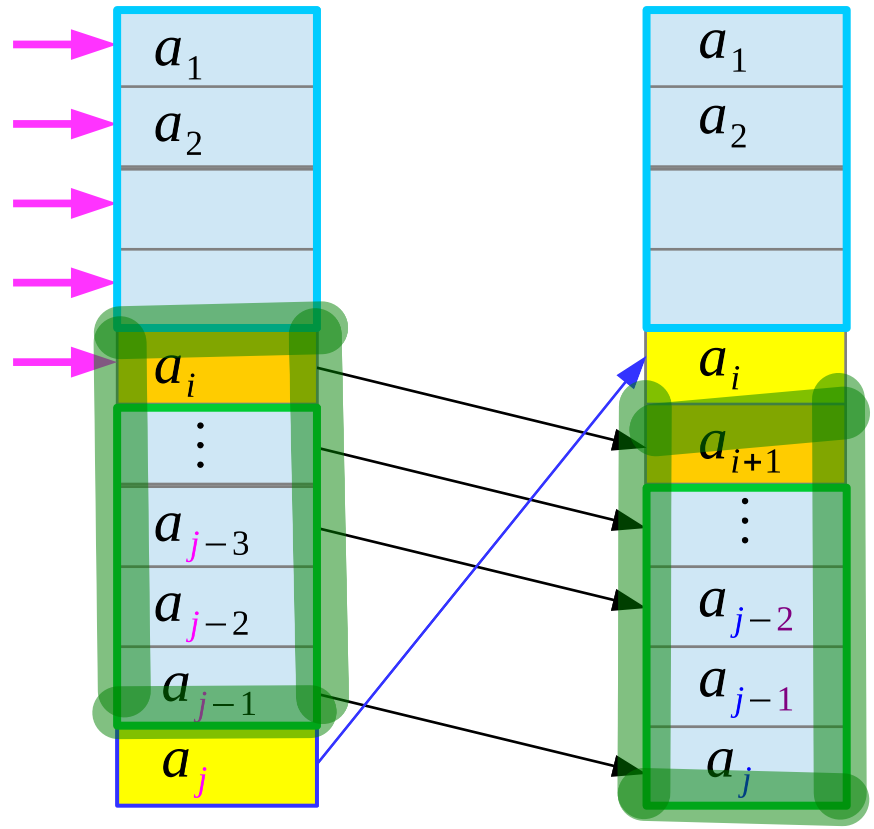
# Nested loop i – finding out of order $\mathbf{a}_i$

```
i := 1
while a_j > a_i
    i := i + 1
```

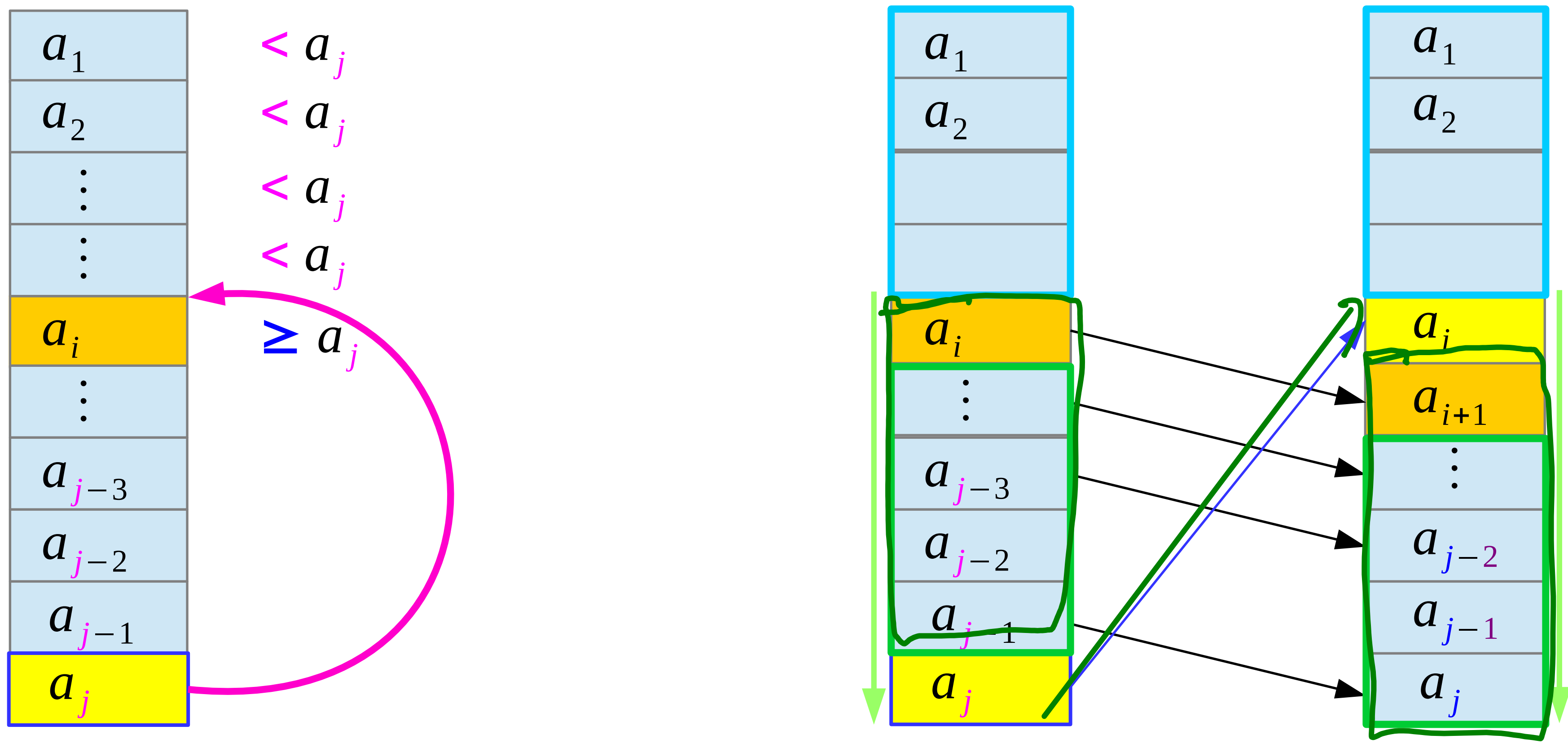
If $a_i$ < $a_j$ increment I

If $a_i$ >= $a_j$ break the loop

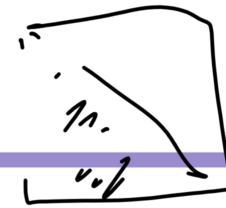
$a_i$ **is the 1st one that is greater than** $a_j$

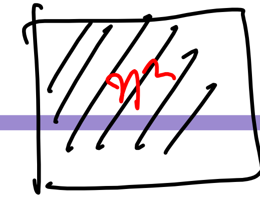


# Nested loop i – inserting $\mathbf{a_i}$ at the correct position

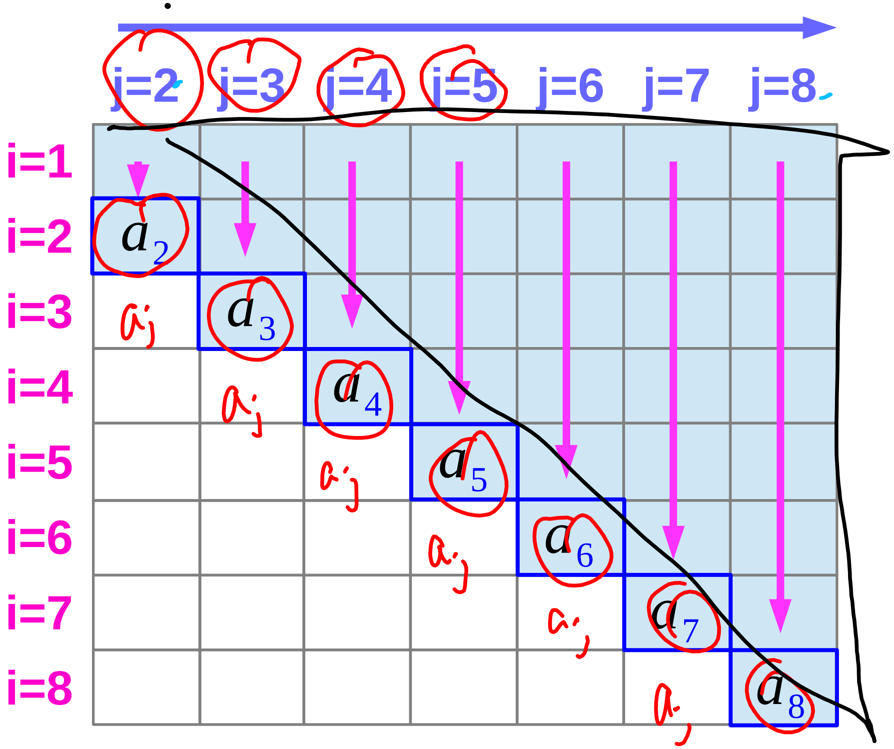


# Nested loop iterations



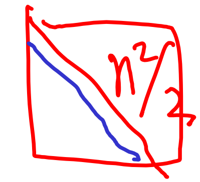



insertion $a_j$




**for** j := 2 **to** n
  i := 1
  **while** $a_j > a_i$
    i := i + 1
  m := $a_j$
  **for** k := 0 **to** j – i–1
    $a_{j-k} = a_{j-k-1}$
  $a_i$ := m

$i < j$




$\frac{n^2}{2} + n$

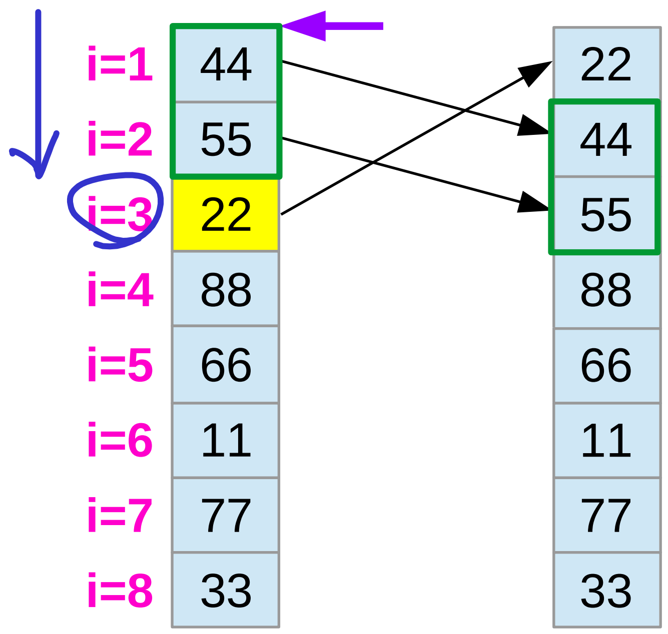
$\Theta(n^2)$

# Step j=2

$a_j = a_2$

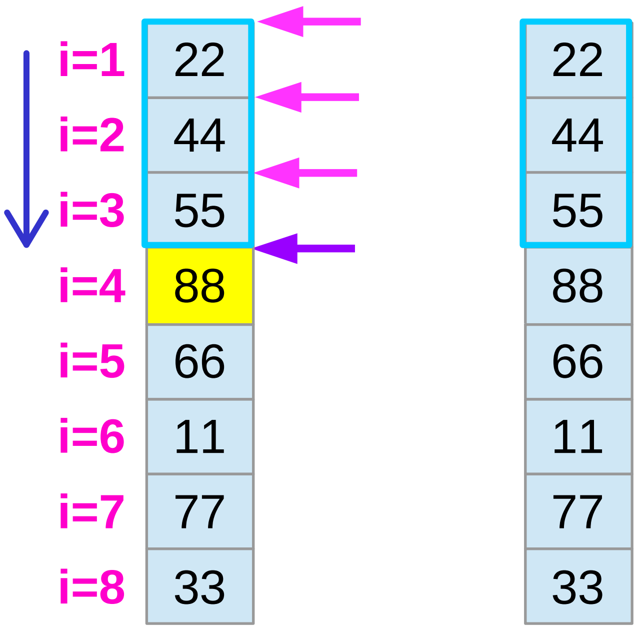
| | | |
|---|---|---|
| i=1 | 44 | 44 |
| i=2 | 55 | 55 |
| i=3 | 22 | 22 |
| i=4 | 88 | 88 |
| i=5 | 66 | 66 |
| i=6 | 11 | 11 |
| i=7 | 77 | 77 |
| i=8 | 33 | 33 |

# Step j=3 $a_j = a_3$



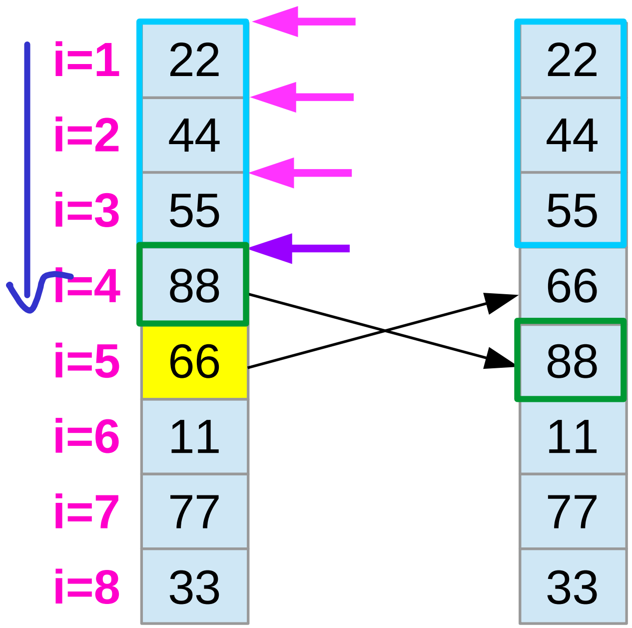
# Step j=4

$a_j = a_4$



# Step j=5

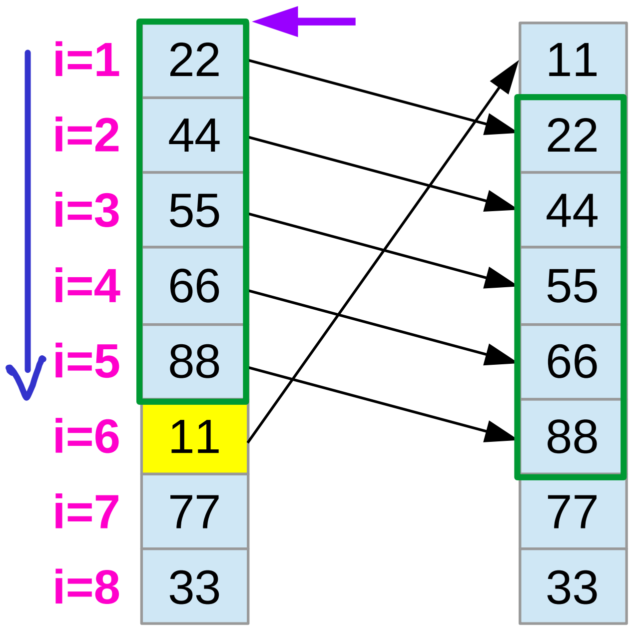
$a_j = a_J$




$\boxed{i < j}$

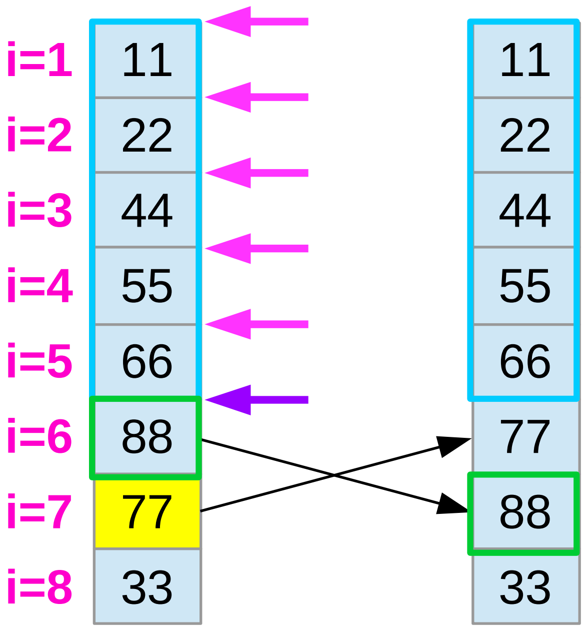
# Step j=6

$a_j = a_6$



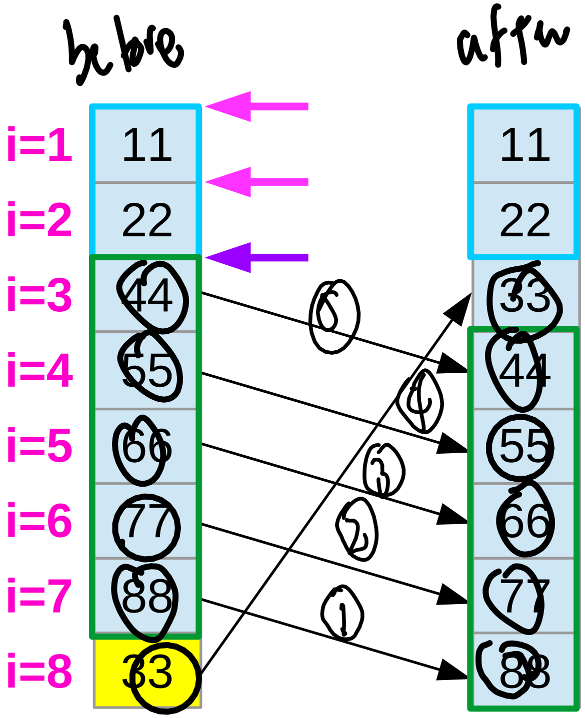
# Step j=7

$a_j = a_n$



# Step j=8

$a_j = a_8$




$k = 0, 1, 2, \ldots$

# Nested loop iterations

## References


[1] http://en.wikipedia.org/
[2]

# Algorithms – Binary Search (1D)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

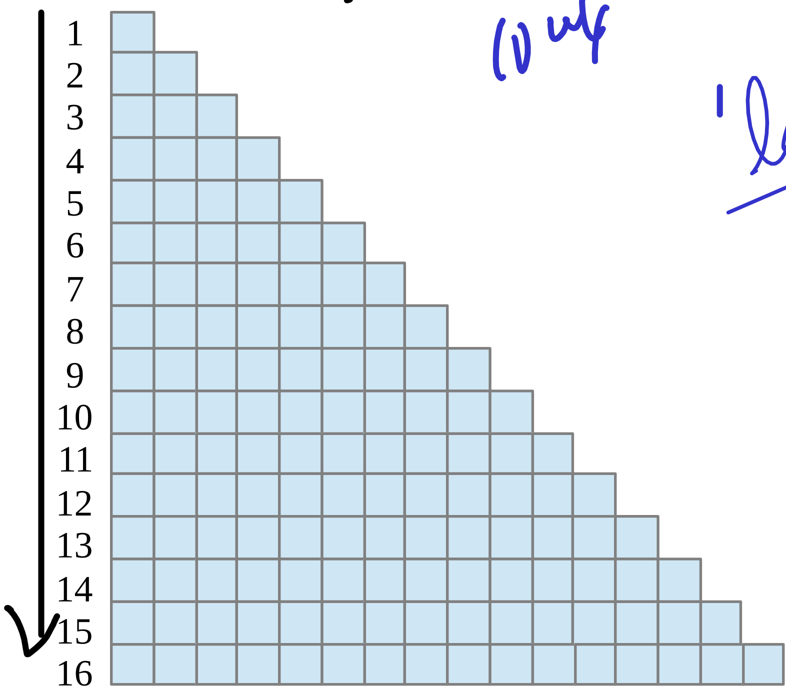
This document was produced by using LibreOffice and Octave.

# Θ(n) vs. Θ(log n)

linear

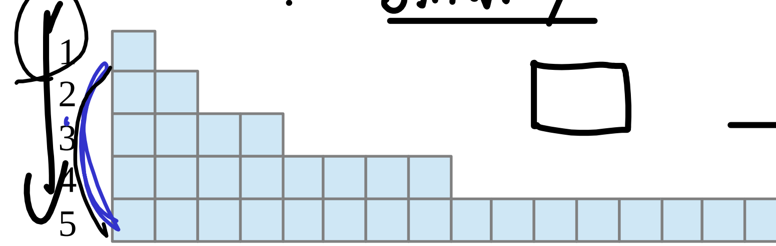
$\log_2 2^4 = 4$

binary

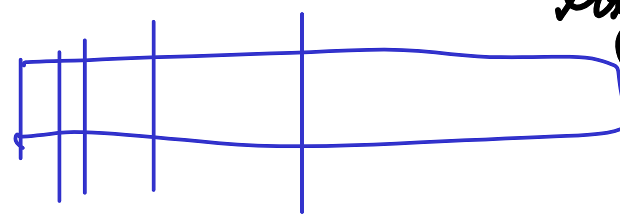


$\log_2 16$

4



$\log_2 2^{10} = 10$



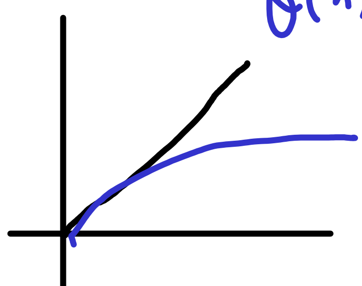
$\boxed{2^{10} = 1024}$

1024

Θ(n)

Θ(log n)

11



https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

# Linear Search Algorithm

$\Theta(n)$

**procedure** linear search(x : integer, $a_1, \ldots, a_n$: distinct integers)
i := 1
**while** (i ≤ **n** and x ≠ $a_i$)
    i := i + 1
**if** (i ≤ **n**) **then** location := i
**else** location := 0
**return** location
{location is the subscript of the term that equals x, or is 0 if x is not found}

$i > n$

3

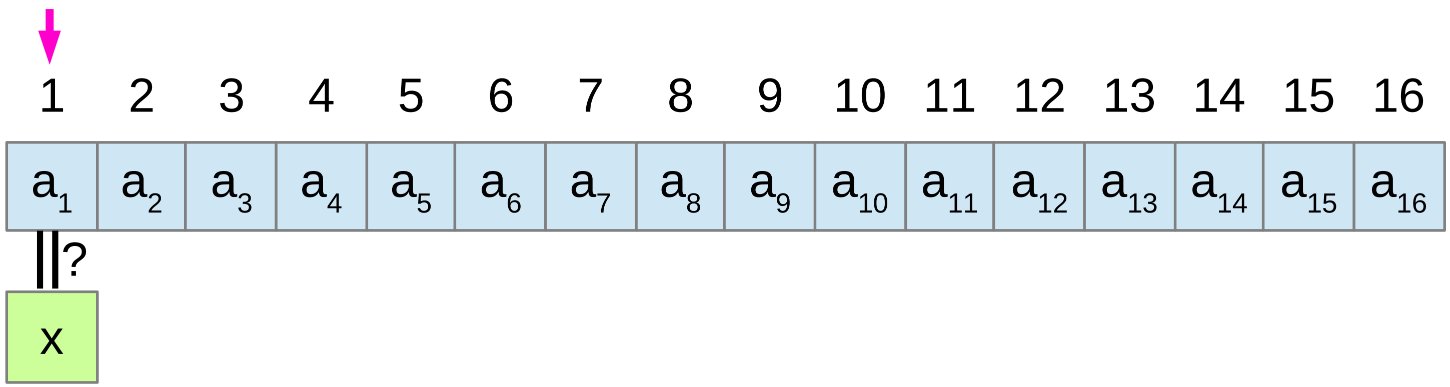
$1 \le i \le n$

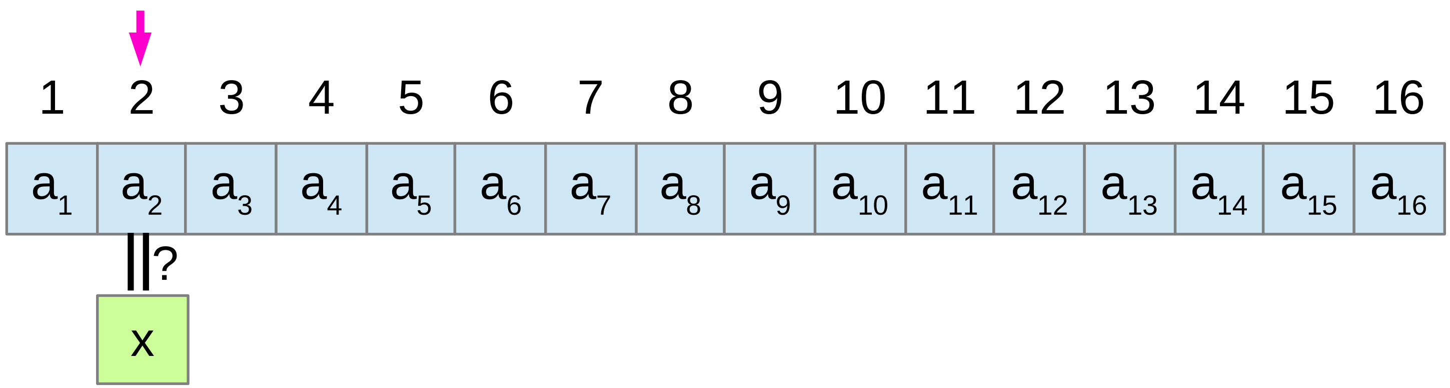
1 2 3 5 7 9

1 2 3 4 5 6

$1 \le i \le n$

# i=1 and i=2


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
$a_1$ $a_2$ $a_3$ $a_4$ $a_5$ $a_6$ $a_7$ $a_8$ $a_9$ $a_{10}$ $a_{11}$ $a_{12}$ $a_{13}$ $a_{14}$ $a_{15}$ $a_{16}$
=?
x


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
$a_1$ $a_2$ $a_3$ $a_4$ $a_5$ $a_6$ $a_7$ $a_8$ $a_9$ $a_{10}$ $a_{11}$ $a_{12}$ $a_{13}$ $a_{14}$ $a_{15}$ $a_{16}$
=?
x

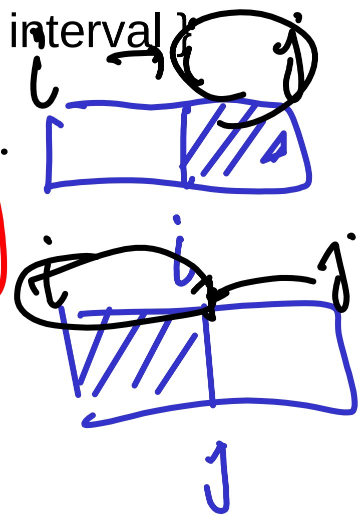
# Best and Worst Cases

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ | $a_{14}$ | $a_{15}$ | $a_{16}$ |

| i = | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ | $a_{14}$ | $a_{15}$ | x | $\parallel$ x |

# Binary Search Algorithm

**procedure** binary search(x : integer, $a_1, \ldots, a_n$ : increasing integers)
i := 1 { i is left endpoint of search interval }
j := n { j is right endpoint of search interval }
**while** (i < j)
    m := **floor**((i+j)/2)
    **if** (x > $a_m$) **then** i := m + 1
    **else** j := m
**if** (x = $a_i$) **then** location := i
**else** location := 0
**return** location
{location is the subscript of the term that equals x, or is 0 if x is not found}

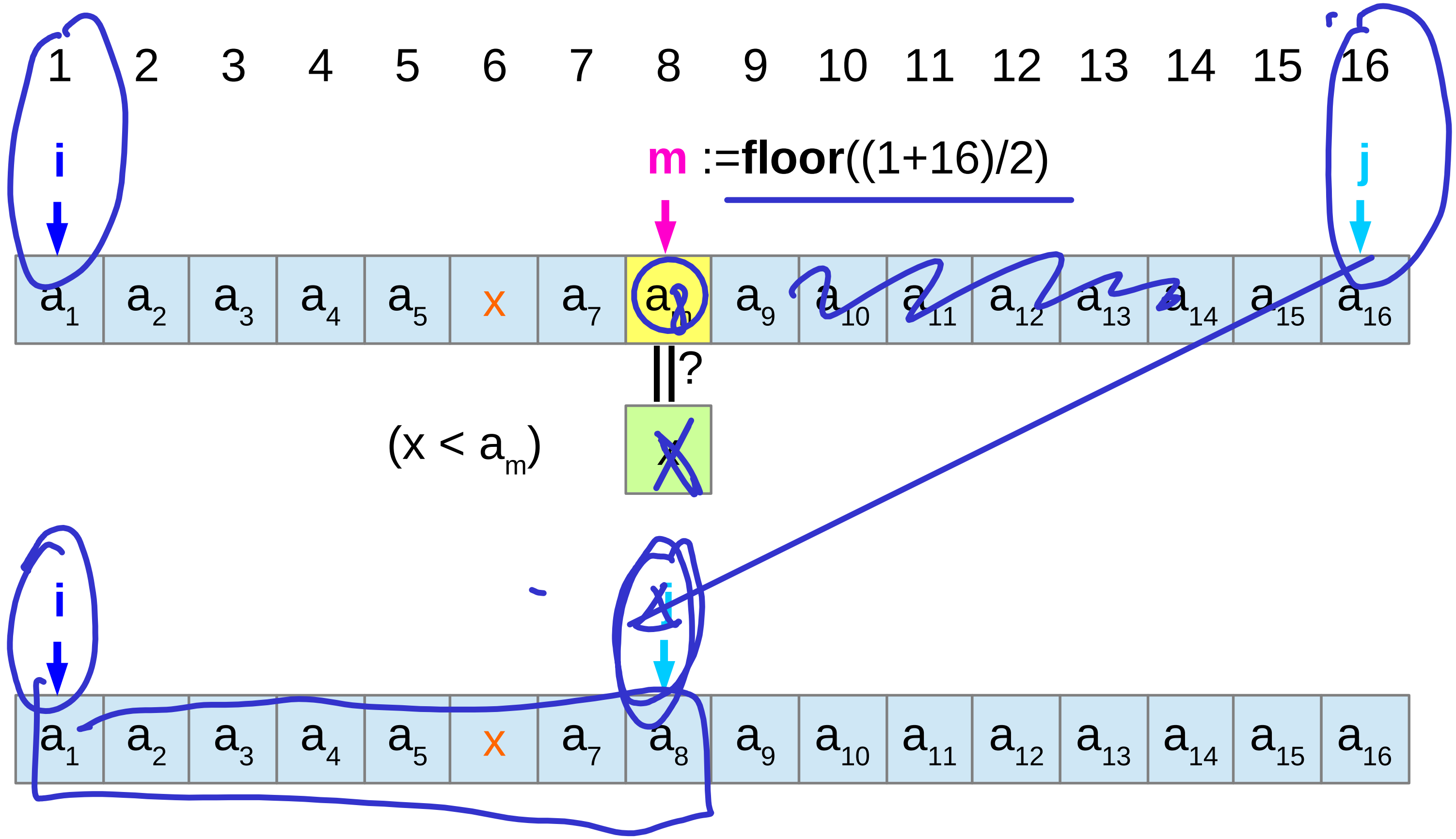


# Increasing Order Assumption

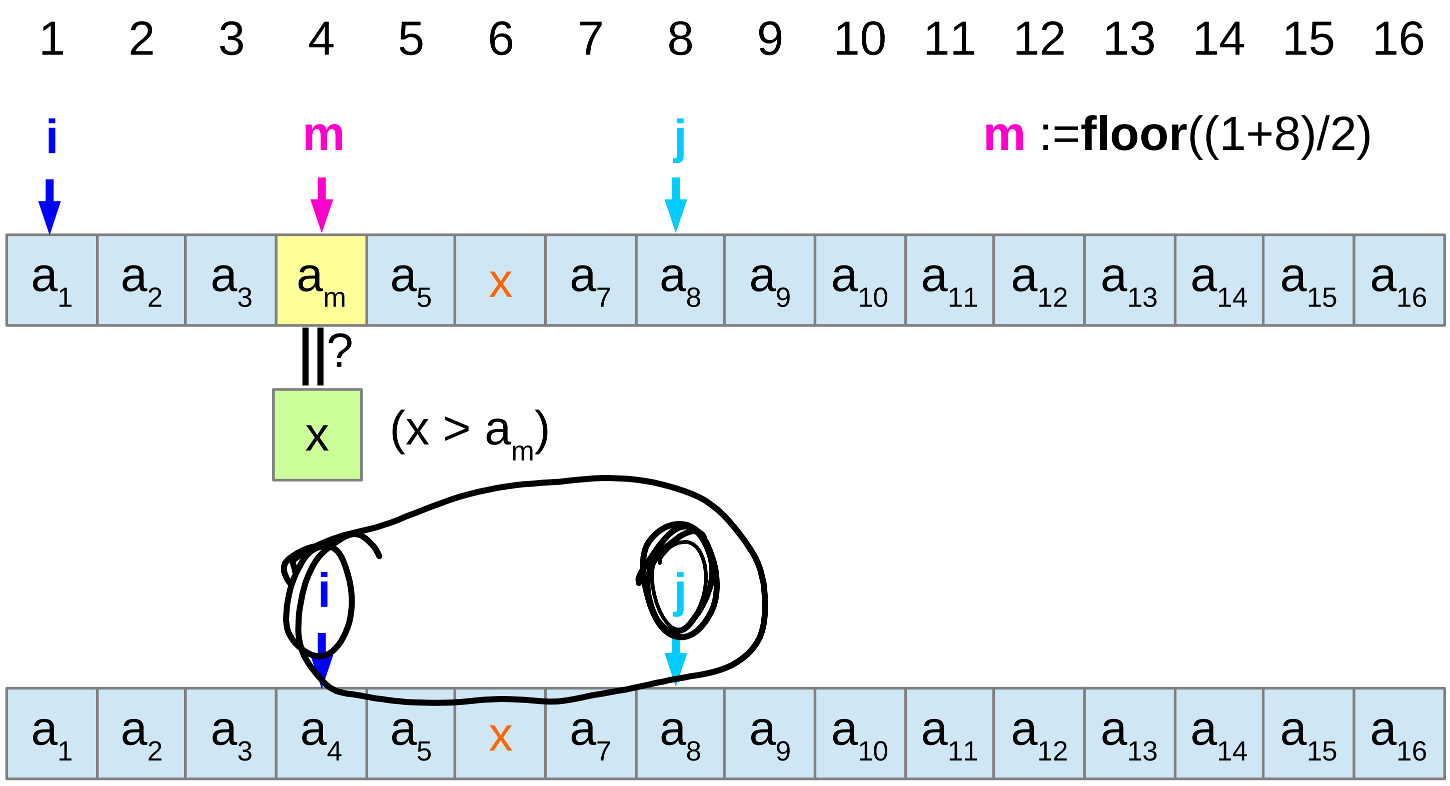
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ | $a_{14}$ | $a_{15}$ | $a_{16}$ |

$$a_1 \le a_2 \le a_3 \le a_4 \le a_5 \le a_6 \le a_7 \le a_8 \le a_9 \le a_{10} \le a_{11} \le a_{12} \le a_{13} \le a_{14} \le a_{15} \le a_{16}$$

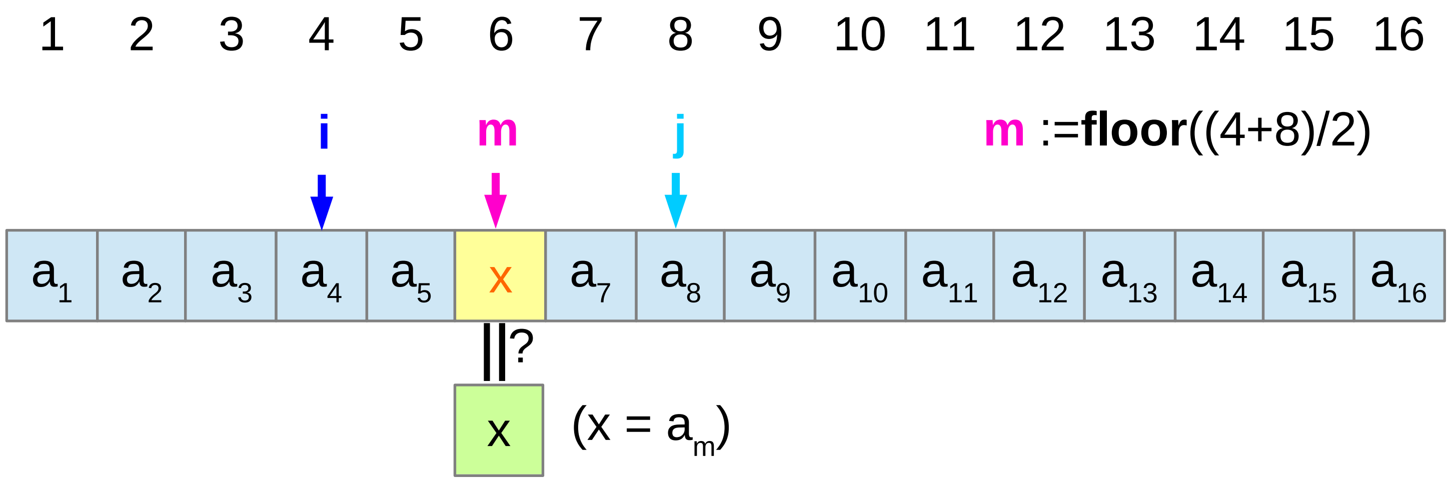
# i=1


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
i
j
m :=**floor**((1+16)/2)
a1 a2 a3 a4 a5 x a7 am a9 a10 a11 a12 a13 a14 a15 a16
=?
(x < am)
x
i
j
a1 a2 a3 a4 a5 x a7 a8 a9 a10 a11 a12 a13 a14 a15 a16

# i=2

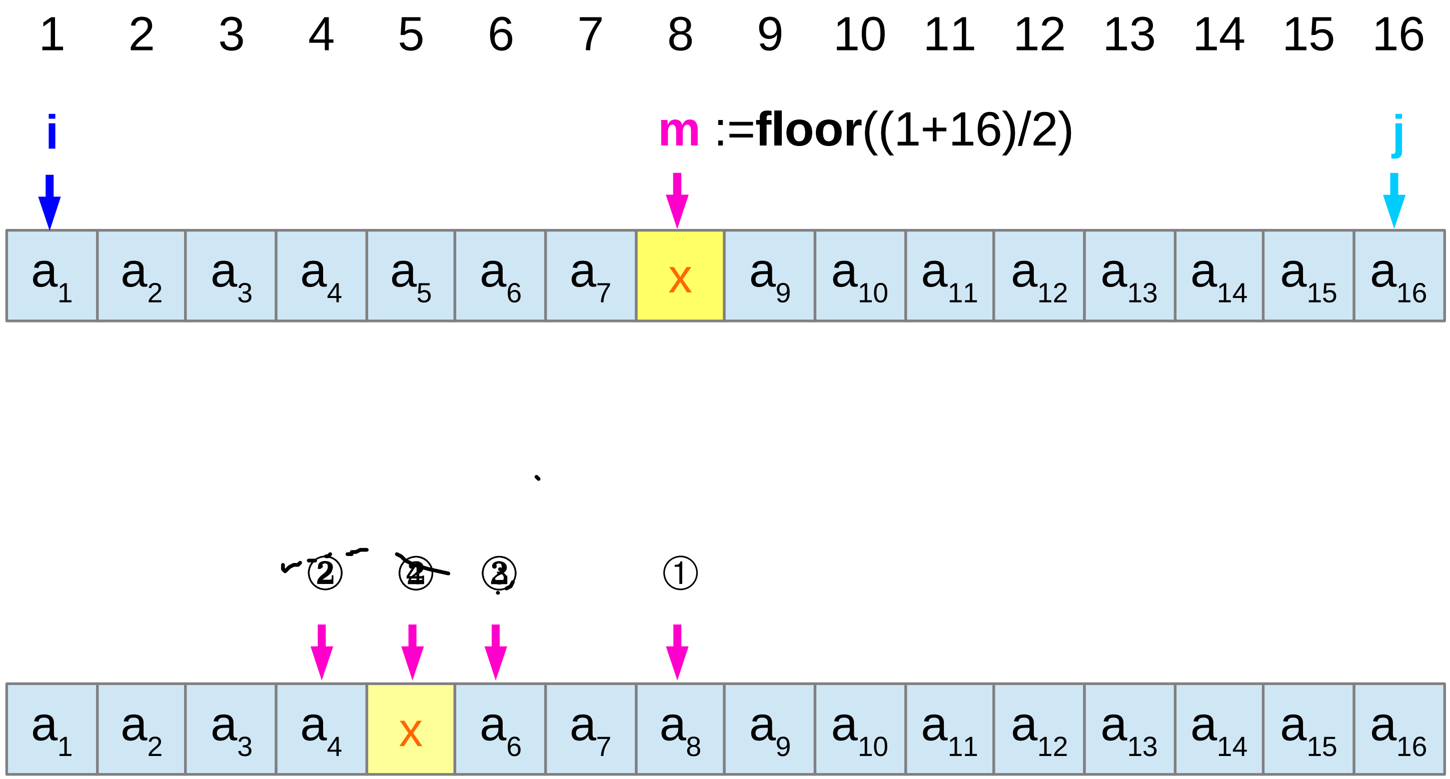

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
i
m
j
m :=floor((1+8)/2)
a1 a2 a3 am a5 x a7 a8 a9 a10 a11 a12 a13 a14 a15 a16
||?
x
(x > am)
i
j
a1 a2 a3 a4 a5 x a7 a8 a9 a10 a11 a12 a13 a14 a15 a16

# i=3



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
i
m
j
m :=**floor**((4+8)/2)
$a_1$ $a_2$ $a_3$ $a_4$ $a_5$ x $a_7$ $a_8$ $a_9$ $a_{10}$ $a_{11}$ $a_{12}$ $a_{13}$ $a_{14}$ $a_{15}$ $a_{16}$
||?
x
$(x = a_m)$

# Best and Worst Cases



# Increasing Order

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| i ↓ | | | | | | | | | | | | | | | j ↓ |
| $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | x | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ | $a_{14}$ | $a_{15}$ | $a_{16}$ |

## References

[1] http://en.wikipedia.org/
[2]

# The Complexity of Algorithms (3A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

# Complexity Analysis

- to compare algorithms at the idea level
  ignoring the low level details
- To measure how fast a program is
- To explain how an algorithm behaves
  as the input grows larger

https://discrete.gr/complexity/

# Counting Instructions

| | |
|---|---|
| • Assigning a value to a variable | x= 100; |
| • Accessing a value of a particular array element | A[i] |
| • Comparing two values | (x > y) |
| • Incrementing a value | i++ |
| • Basic arithmetic operations | +, –, *, / |
| • Branching is not counted | if else |

https://discrete.gr/complexity/

# Asymptotic Behavior

- avoiding tedious instruction counting
- eliminate all the minor details
- focusing how algorithms behaves when treated badly
- drop all the terms that grow slowly
- only keep the ones that grow fast as **n** becomes larger

https://discrete.gr/complexity/

# Finding the Maximum

```
M = A[0];
for (i=0; i<n; ++i) {
    if (A[i] >= M) {
        M = A[i];
    }
}
```

// M is set to the 1st element

// if the (i+1)th element is greater than M,
// M is set to that element (new maximum value)

```
int A[n];     // n element integer array A
int M;        // the current maximum value found so far
              // set to the 1st element, initially
```

https://discrete.gr/complexity/

# Worst and Best Cases

int A[4];

| | |
|---|---|
| i=0 | A[0] |
| i=1 | A[1] |
| i=2 | A[2] |
| i=3 | A[3] |

Case 1:
Worst Case

| | | |
|---|---|---|
| A[0]=1 | → | M=1 |
| A[1]=2 | → | M=2 |
| A[2]=3 | → | M=3 |
| A[3]=4 | → | M=4 |

Case 2:
Best Case

| | | |
|---|---|---|
| A[0]=4 | → | M=4 |
| A[1]=3 | | |
| A[2]=2 | | |
| A[3]=1 | | |

```
for (i=0; i<n; ++i) {
    if (A[i] >= M) {       // always n comparisons
        M = A[i];          // the updating of M depends on the data
    }                      // minimum 1 update, maximum n updates
```

https://discrete.gr/complexity/

# Assignment

```
M = A[0];                    // 2 instructions
for (i=0; i<n; ++i) {
        if (A[i] >= M) {
                M = A[i];
        }
}
```

A[0] – **1** instruction

M = – **1** instruction

https://discrete.gr/complexity/

# Loop instructions

```
for (i=0; i<n; ++i) {
    if (A[i] >= M) {
        M = A[i];
    }
```

Initialization * **1**

| | |
|---|---|
| i=0 | : **1** instruction |
| i<n | : **1** instruction |

Update * **n**

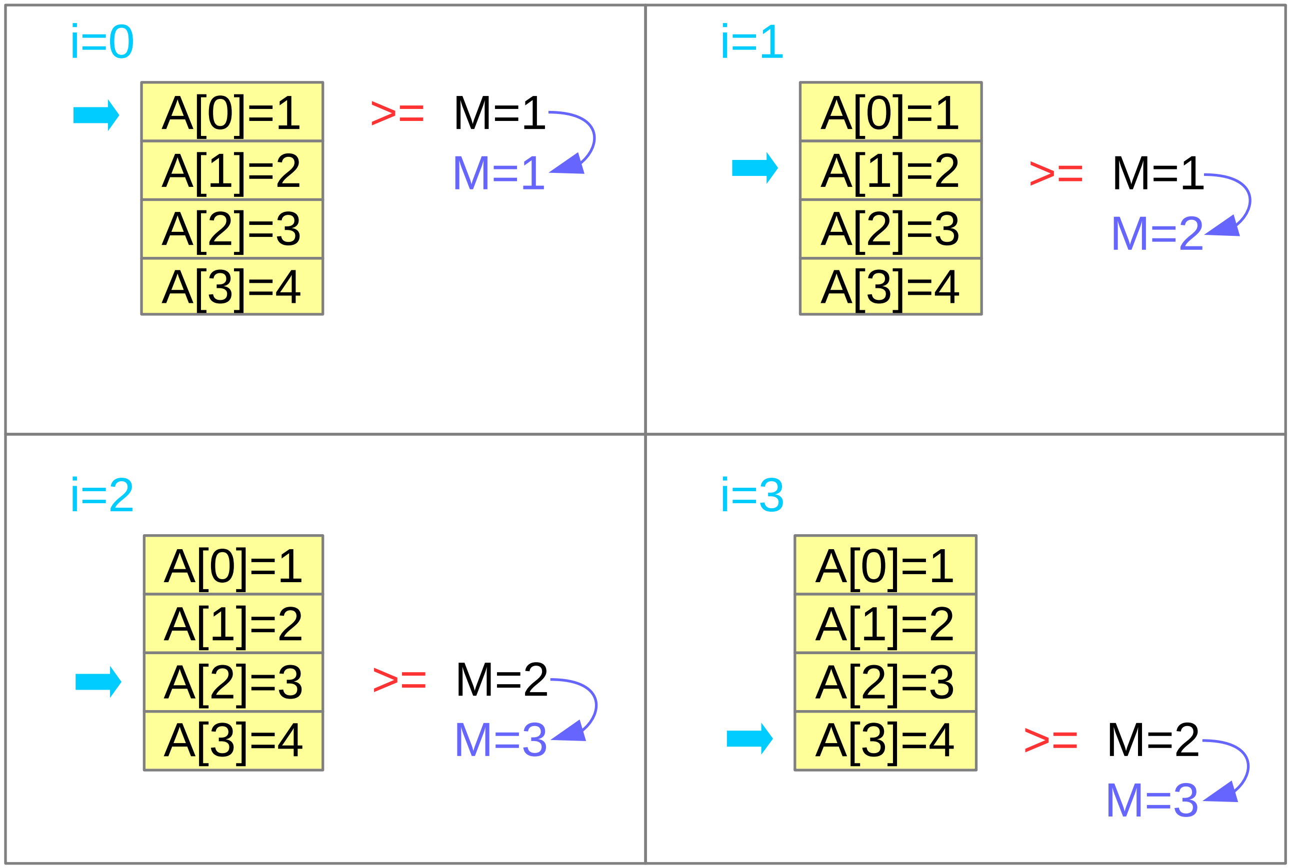
| | |
|---|---|
| ++i | : **1** instruction |
| i<n | : **1** instruction |

Loop body * **n**

| | | | |
|---|---|---|---|
| A[i] | : **1** instruction | **n** | always |
| >= M | : **1** instruction | | |
| A[i] | : **1** instruction | **1 ~ n** | depending on the comparison |
| M= | : **1** instruction | | |

https://discrete.gr/complexity/

# Worst case examples




```
for (i=0; i<n; ++i) {
    if (A[i] >= M) {
        M = A[i];
    }
```

2**n** + 2**n** = 4**n**

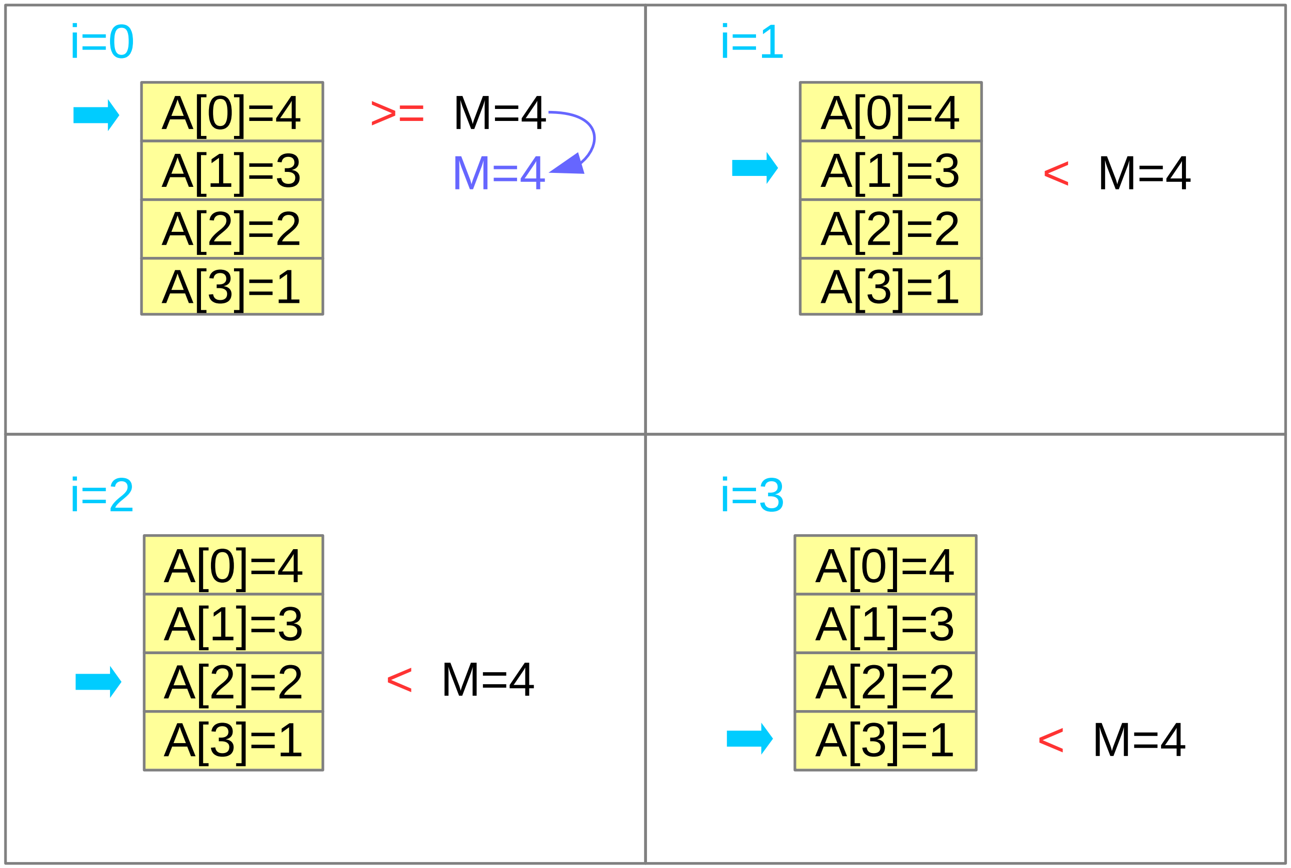
Instructions

**n** comparisons

**n** updates

https://discrete.gr/complexity/

# Best case examples




```
for (i=0; i<n; ++i) {
    if (A[i] >= M) {
        M = A[i];
    }
```

2**n** + 2

Instructions

**n** comparisons

**1** update

https://discrete.gr/complexity/

# Asymptotic behavior

| | | |
|---|---|---|
| M = A[0]; - - - - - - - - - - - - - - - - | 2 | instructions |
| for (i=0; i<**n**; ++i) { - - - - - - - - | 2 + 2**n** | instructions (init + update) |
|     if (A[i] >= M) { - - - - - - - - | 2**n** | instructions |
|         M = A[i]; - - - - - - - - - | 2 ~ 2**n** | instructions |
|     } | | |
| } | | |

$$f(n) = \begin{cases} 6n+4 & \text{instructions for the worst case} \\ 4n+6 & \text{instruction for the best case} \end{cases}$$

$f(n) = O(n)$

$f(n) = \Omega(n)$

$f(n) = \Theta(n)$

https://discrete.gr/complexity/

# Θ(n) codes

$\Theta(n)$

```
// Here c is a positive integer constant
for (i = 1; i <= n; i += c) {
    // some O(1) expressions
}

for (int i = n; i > 0; i -= c) {
    // some O(1) expressions
}
```

$1, 2, 3, \ldots\ldots, n$

$n = \Theta(n)$

for( i=1 ; i≤n ; i+=2)

$1, 3, 5, 7, \ldots, n$

$\approx \frac{n}{2} = \Theta(n)$

for (i=1 ; i<=n ; i+=3)

1 2 3 | 4 5 6 | 7 8 9 | 10 11 12 | ... | 7

https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

$\approx \frac{n}{3}$ 13 $= \Theta(n)$

for (i=n; i>0; i-=3)

i = n  n-1  n-2 .... 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

n

$\frac{n}{3} = \theta(n)$

$\frac{n}{2} = \theta(n)$

$x^2+2x+1 = \Theta(x^2)$ $\quad$ $n = \Theta(n)$

$6x^3+2x^2+1 = \Theta(x^3)$ $\quad$ $\frac{1}{2}n = \Theta(n)$

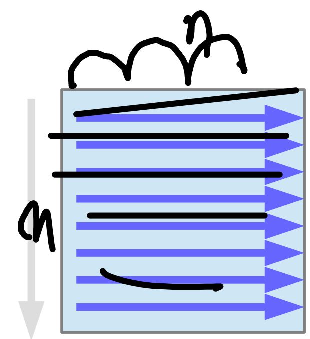
$\approx 6x^3.$

# $O(n^2)$ codes

```
for (i = 1; i <=n; i += c) {
   for (j = 1; j <=n; j += c) {
      // some O(1) expressions
   }
}
```

$\Theta(n^2)$ $\left(\frac{n^2}{4}\right)$

$\Theta(n^2)$

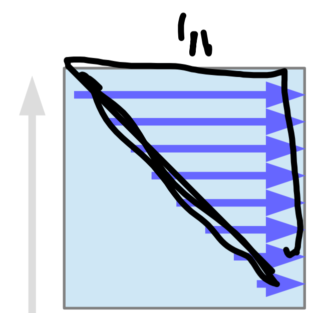


$\left(\frac{n}{2}\right)\left(\frac{n}{2}\right) = \frac{n^2}{4}$

$n^2$

```
for (i = n; i > 0; i += c) {
   for ( j = i+1; j <=n; j += c) {
      // some O(1) expressions
}
```

$\Theta(n^2)$



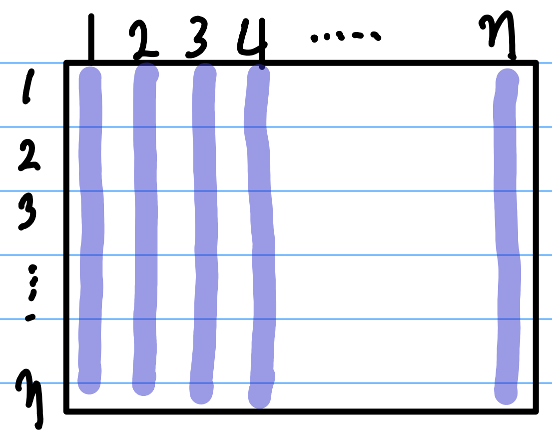
$\frac{n^2}{2} - n$

for(i=n; i>0; i+=1)

for(j=i+1, j<=n; j+=1)

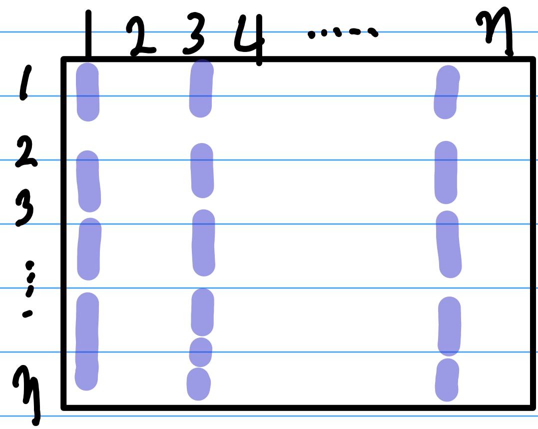
https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

```
for (i=1; i<=n; i+=1)
    for (j=1; j<=n; j+=1)
```



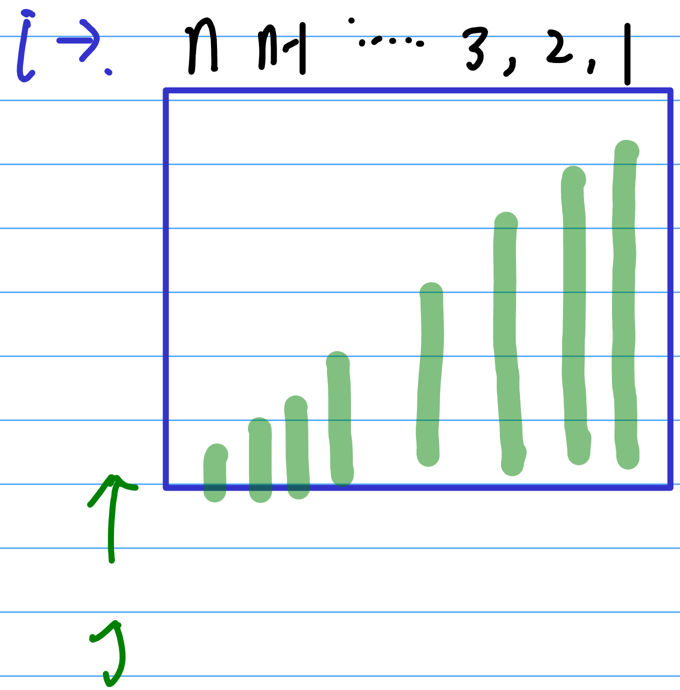

$$n \cdot n = n^2 = \Theta(n^2)$$

```
for (i=1; i<=n; i+=2)
    for (j=1; j<=n; j+=2)
```




$$\left(\frac{n}{2}\right)\cdot\left(\frac{n}{2}\right) = \frac{n^2}{4} = \Theta(n^2)$$

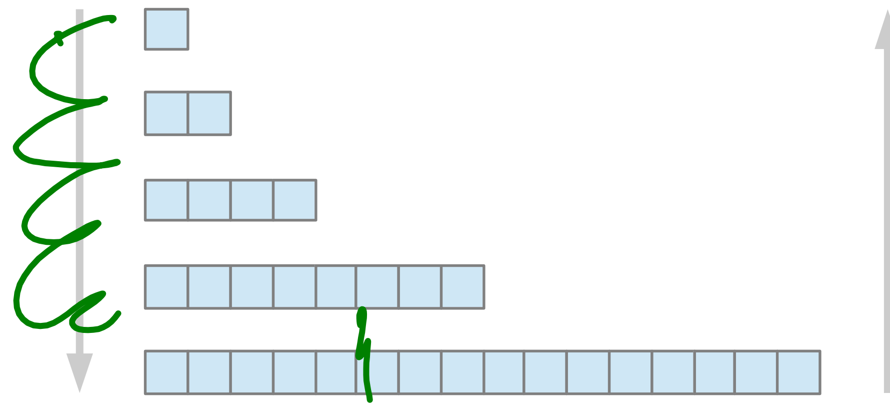
```
for(i=n; i>0; i-=1)
    for(j=i+1, j<=n; j+=1)
```




# O(log **n**) codes

$\Theta(\log n)$

```
for (int i = 1; i <=n; i *= c) {
   // some O(1) expressions
}
for (int i = n; i > 0; i /= c) {
   // some O(1) expressions
}
```



https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

$i += 1 \qquad i = i+1$

$i += 2 \qquad i = i+2$

$i *= 2 \qquad i = \underline{i*2}$

for (i=1; i<=1024; i*=2)

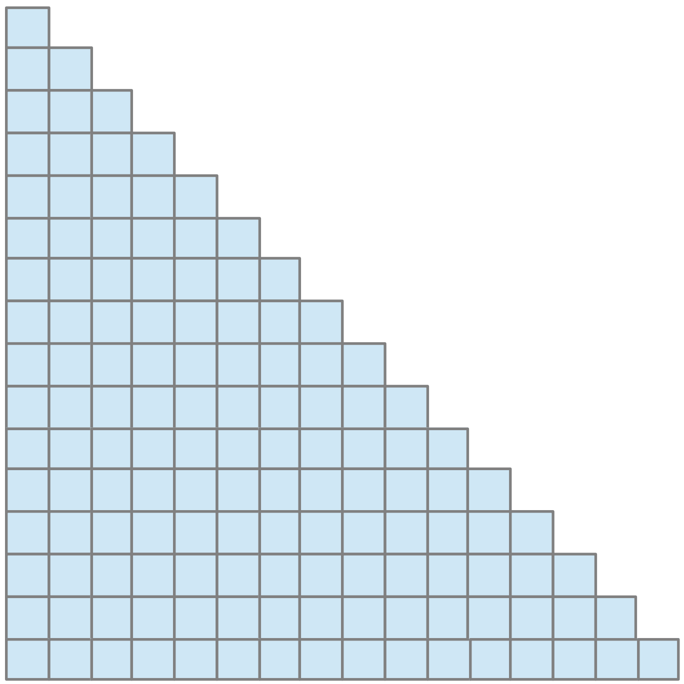
1 2 3 4 5 6 7 8 9 10 11

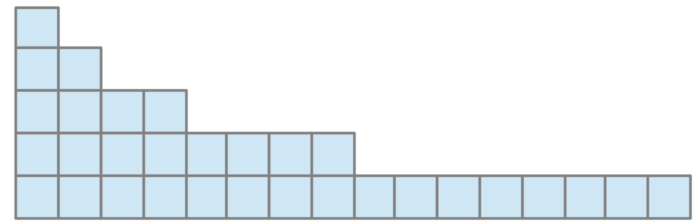
$i = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024$

$2^0 \quad 2^1 \quad 2^2 \quad 2^3 \quad 2^4 \quad 2^5 \quad 2^6 \quad 2^7 \quad 2^8 \quad 2^9 \quad 2^{10}$

$\log_2 2^x + 1$

# O(**n**) vs. O(log **n**)





https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

# O(log **n**) codes

```
// Here c is a constant greater than 1
for (int i = 2; i <=n; i = pow(i, c)) {          // i = i^c
   // some O(1) expressions
}
//Here fun is sqrt or cuberoot or any other constant root
for (int i = n; i > 0; i = fun(i)) {             // i = i^(1/c)
   // some O(1) expressions
}
```

$i = i^2, i = i^3$

https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

# O(log log **n**) codes

```
// Here c is a constant greater than 1
for (int i = 2; i <=n; i = pow(i, c)) {          // i = i^c
   // some O(1) expressions
}
//Here fun is sqrt or cuberoot or any other constant root
for (int i = n; i > 0; i = fun(i)) {             // i = i^(1/c)
   // some O(1) expressions
}
```

$i = i^2 \quad (2, 2^2, 2^4, 2^8, 2^{16}, \cdots)$

$i = i^{\frac{1}{2}} \quad (n, n^{\frac{1}{2}}, n^{\frac{1}{4}}, n^{\frac{1}{8}}, n^{\frac{1}{16}}, \cdots)$

https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

# O(log log **n**) codes

```
// Here c is a constant greater than 1
for (int i = 2; i <=n; i = pow(i, c)) {            // i = i^c
   // some O(1) expressions
}
//Here fun is sqrt or cuberoot or any other constant root
for (int i = n; i > 0; i = fun(i)) {              // i = i^(1/c)
   // some O(1) expressions
}
```

$i = i^2 \quad (2, 2^2, 2^4, 2^8, 2^{16}, \cdots)$

$i = i^{\frac{1}{2}} \quad (n, n^{\frac{1}{2}}, n^{\frac{1}{4}}, n^{\frac{1}{8}}, n^{\frac{1}{16}}, \cdots)$

https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

# Some Algorithm Complexities and Examples (1)

O → Θ k Knuth Θ, Θ

**O(1) – Constant Time**
not affected by the input size **n**.

**O(n) – Linear Time**
Proportional to the input size **n**.

**O(log n) – Logarithmic Time**
recursive subdivisions of a problem
binary search algorithm

**O(n log n) – Linearithmic Time**
Recursive subdivisions of a problem and then merge them
merge sort algorithm.

https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

# Some Algorithm Complexities and Examples (2)

$n^2$

**O(n2) – Quadratic Time**

bubble sort algorithm

**O(n3) – Cubic Time**

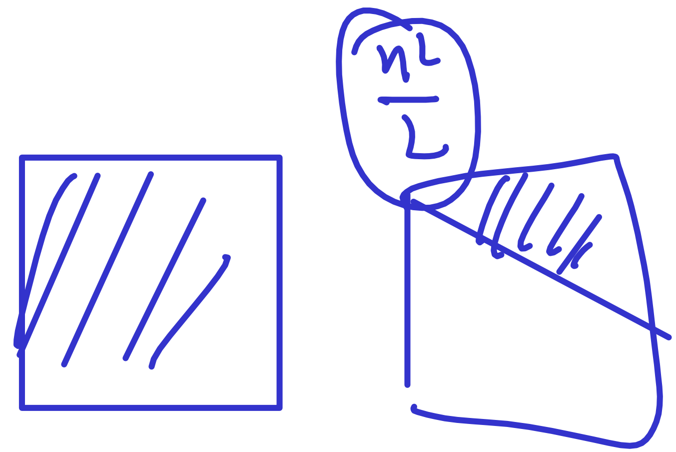
straight forward matrix multiplication

**O(2^n) – Exponential Time**

Tower of Hanoi

**O(n!) – Factorial Time**

Travel Salesman Problem (TSP)




$O(n^2)$

https://stackoverflow.com/questions/11032015/how-to-find-time-complexity-of-an-algorithm

## References


[1] http://en.wikipedia.org/
[2]