Sorting (P.1)

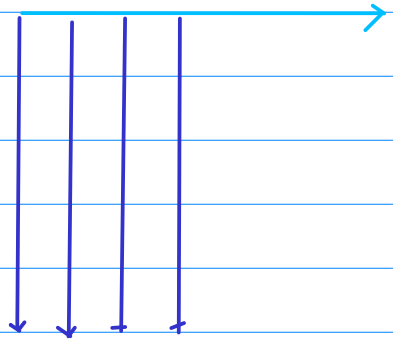
used some pictures and codes from
http://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf
Data Structures and Algorithm Analysis
by Clifford A. Schaffer

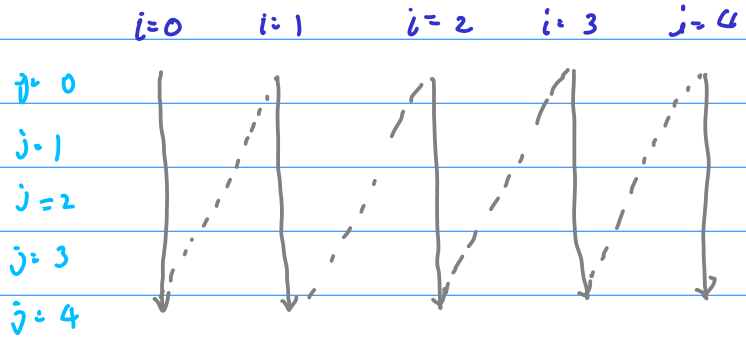
```cpp
#include <stdio.h>

template <class T>
T square(T x) {
  return (x*x);
}


int main(void) {

  int    i2, i=2;
  float  f2, f=3.0;
  double d2, d=4.0;

  i2 = square<int>(i);
  f2 = square<int>(f);
  d2 = square<int>(d);

  printf("i= %d i2= %d \n", i, i2);
  printf("f= %f f2= %f \n", f, f2);
  printf("d= %f d2= %f \n", d, d2);

  return 0;

}
```

typename

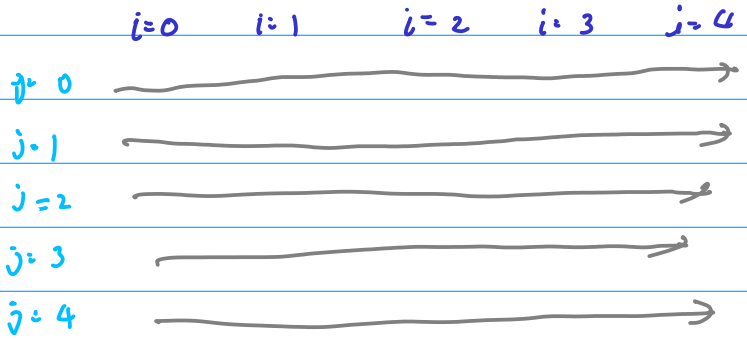
int
float
double

```
// Swap two elements in a generic array
template<typename E>
inline void swap(E A[], int i, int j) {
  E temp = A[i];
  A[i] = A[j];
  A[j] = temp;
}
// Random number generator functions
```



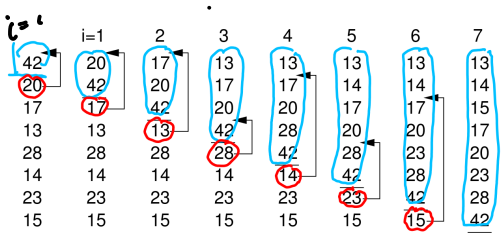



```
for (i=0; i<5; ++i)
  for (j=0; j<5; ++j)
```

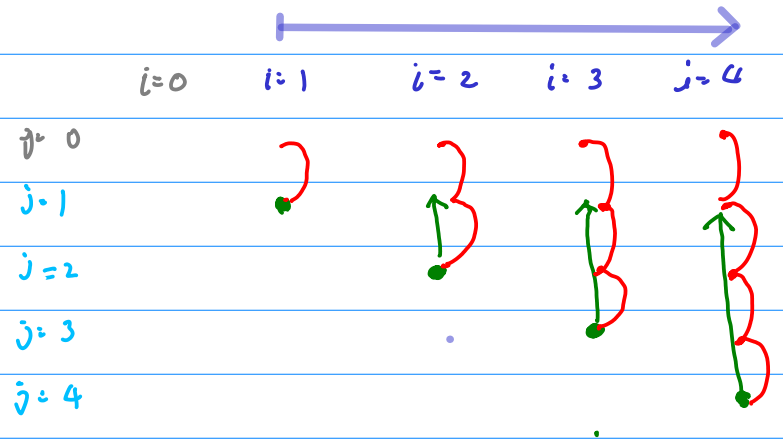



```
for (j=0; j<5; ++j)
  for (i=0; i<5; ++i)
```

Insertion Sort

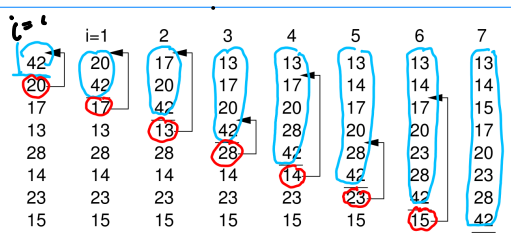



```
template <typename E, typename Comp>
void inssort(E A[], int n) { // Insertion Sort
  for (int i=1; i<n; i++)       // Insert i'th record
    for (int j=i; (j>0) && (Comp::prior(A[j], A[j-1])); j--)
      swap(A, j, j-1);
}
```

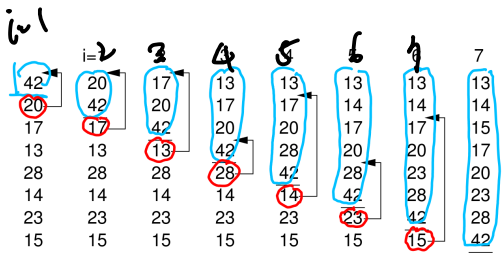



```
for (i=1; i<5; ++i)
  for (j=i; j>0; --j)
    if (A[j] < A[j-1]) swap(A[j], A[j-1])
```

```
for (i=1; i<5; ++i)
  for (j=i; j>0; --j)
    if (A[j] < A[j-1]) swap(A[j], A[j-1])
    else break;
```



	i=1	i=2		i=3			i=4			
		j=2	j=1	j=3	j=2	j=1	j=4	j=3	j=2	
0	42	20	20	17	17	17	13	13	13	13
1	20	42	17	20	20	13	17	17	17	17
2	17	17	42	42	13	20	20	20	28	28
3	13	13	13	13	42	42	42	28	20	20
4	28	28	28	28	28	28	28	42	42	42
5	14	14	14	14	14	14	14	14	14	14
6	23	23	23	23	23	23	23	23	23	23
7	15	15	15	15	15	15	15	15	15	15




```
i=1 j=1
----------------------
i=2 j=2
i=2 j=1
----------------------
i=3 j=3
i=3 j=2
i=3 j=1
----------------------
i=4 j=4
----------------------
i=5 j=5
i=5 j=4
i=5 j=3
i=5 j=2
----------------------
i=6 j=6
i=6 j=5
----------------------
i=7 j=7
i=7 j=6
i=7 j=5
i=7 j=4
i=7 j=3
```

```
for   i = 1 to N
    for   j = 2 to (N-1)
        do {
            Read   A[i, j] ;
            Write  A[i, j] ;
        }
```

(2)

$(N-1) - 2 + 1$

$(N-2) \times 2$ ← Read, Write

실행되는 명령의 횟수

- Read = A
- write A =
- Comp A < B >

$N \times (N-2) \times 2 = 2N^2 - 4N = O(N^2)$

Quadratic time Alg.

```
#define MAX 9000000

for i = 1000 to N-20000
    for j = 2 to MAX
        do {
        Read A[i,j];
        write A[i,j];
    }
```

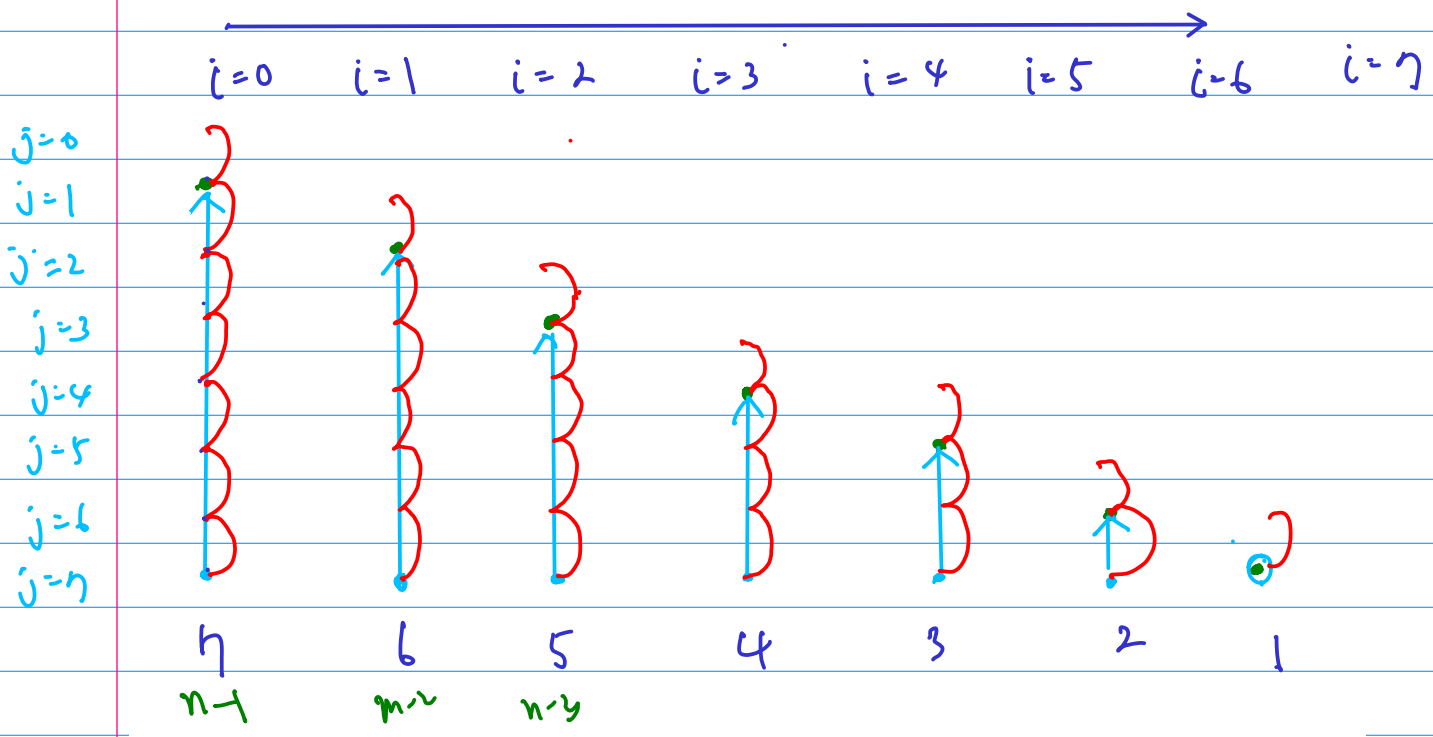
$N - 20000 - 1000$
$= N - 19000$

$9000000 - 2 + 1$
9999999

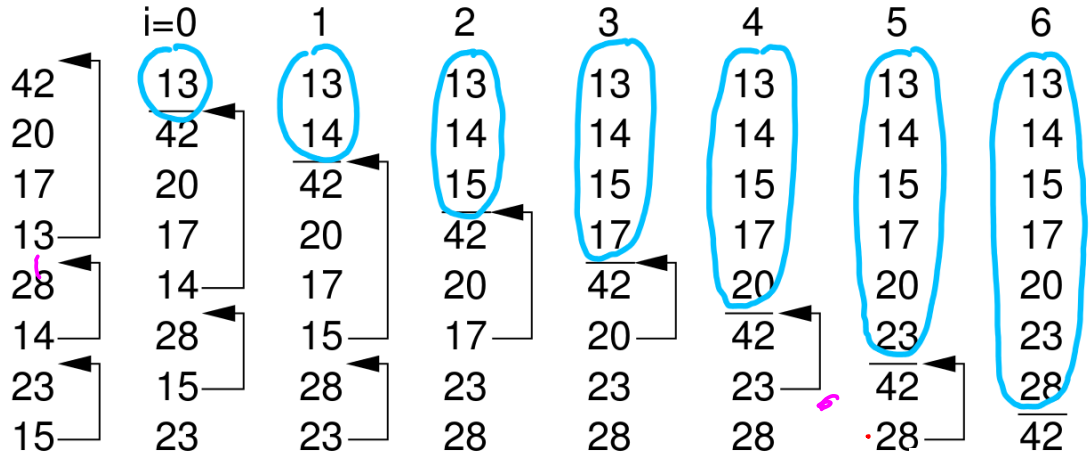
$(N - 19000) \times 9999999 \times 2 = O(N)$

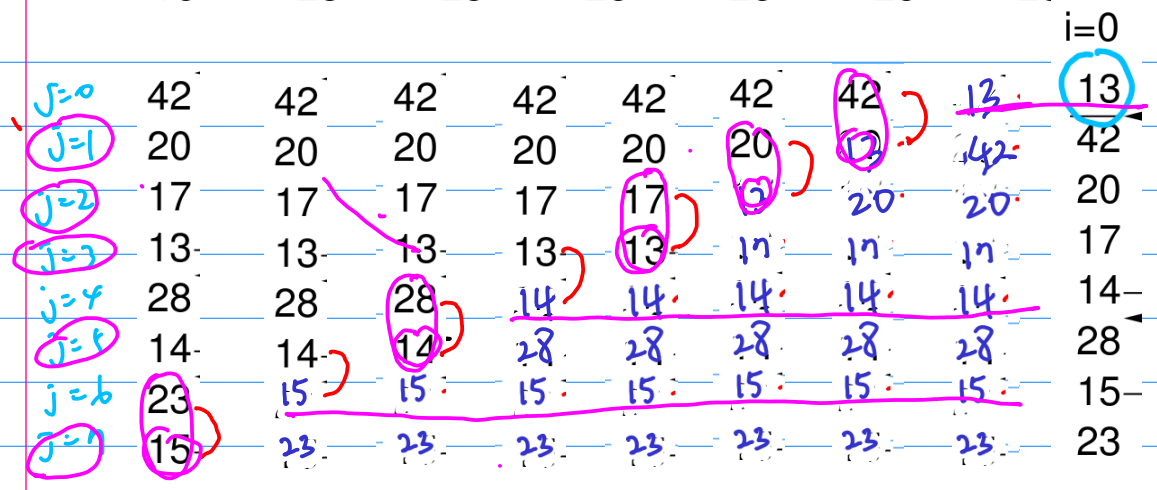
Linear time alg.

```
template <typename E, typename Comp>
void bubsort(E A[], int n) { // Bubble
  for (int i=0; i<n-1; i++)      // Bubl
    for (int j=n-1; j>i; j--)
      if (Comp::prior(A[j], A[j-1]))
        swap(A, j, j-1);
}
```



	i=0	1	2	3	4	5	6
42	13	13	13	13	13	13	13
20	42	14	14	14	14	14	14
17	20	42	15	15	15	15	15
13	17	20	42	17	17	17	17
28	14	17	20	42	20	20	20
14	28	15	17	20	42	23	23
23	15	28	23	23	23	42	28
15	23	23	28	28	28	28	42





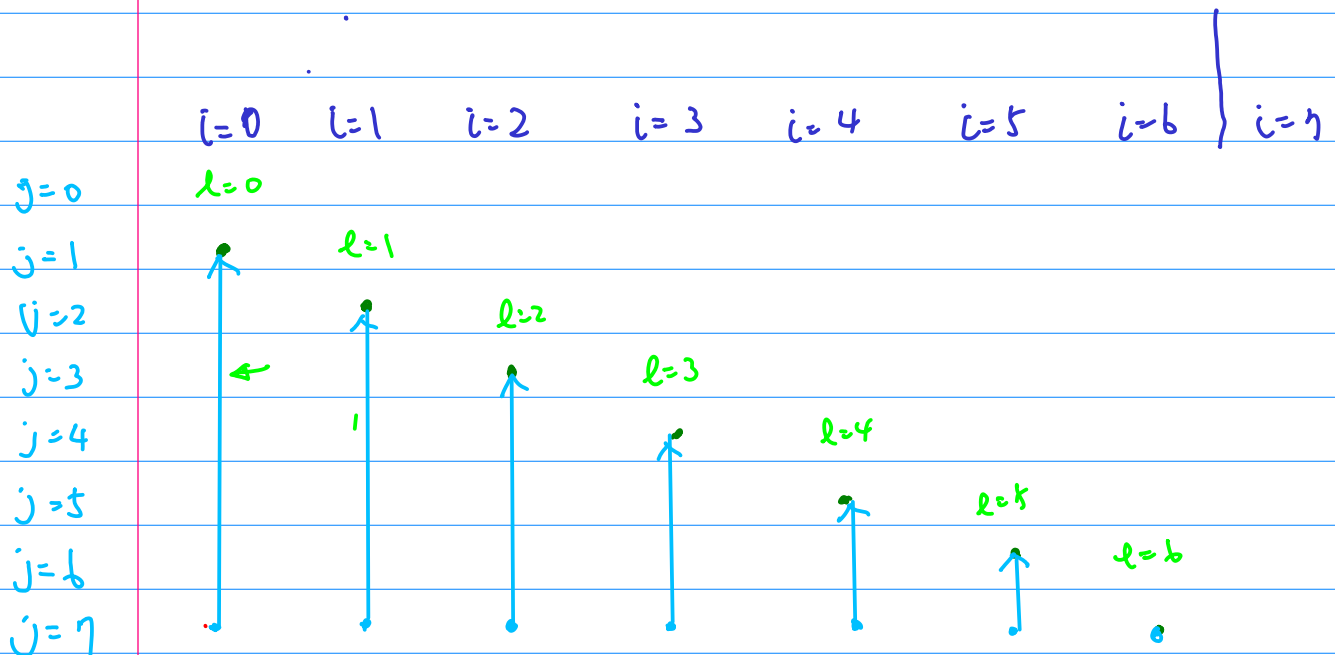

```
i=0 j=7
i=0 j=5
i=0 j=3
i=0 j=2
i=0 j=1
------------
i=1 j=6
i=1 j=4
i=1 j=3
i=1 j=2
------------
i=2 j=7
i=2 j=5
i=2 j=4
i=2 j=3
------------
i=3 j=5
i=3 j=4
------------
i=4 j=5
------------
i=5 j=6
------------
i=6 j=7
------------
```

# of comparisons $A[j] < A[j-1]$

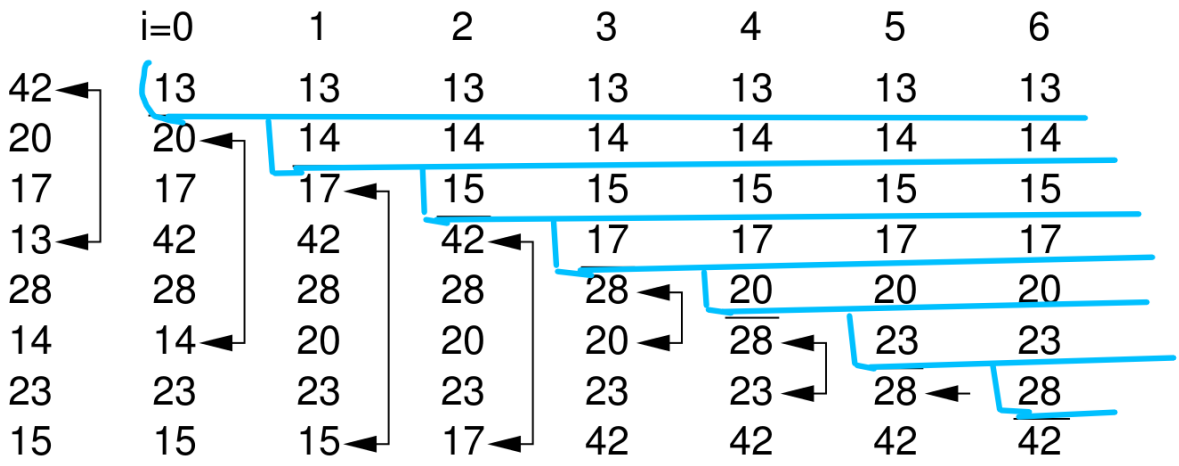
$$(n-1) + (n-2) + (n-3) + \cdots + 3 + 2 + 1$$

$$= \sum_{i=1}^{n-1} i = \frac{(n-1)(n-1+1)}{2} = \frac{n^2-n}{2} = O(n^2)$$

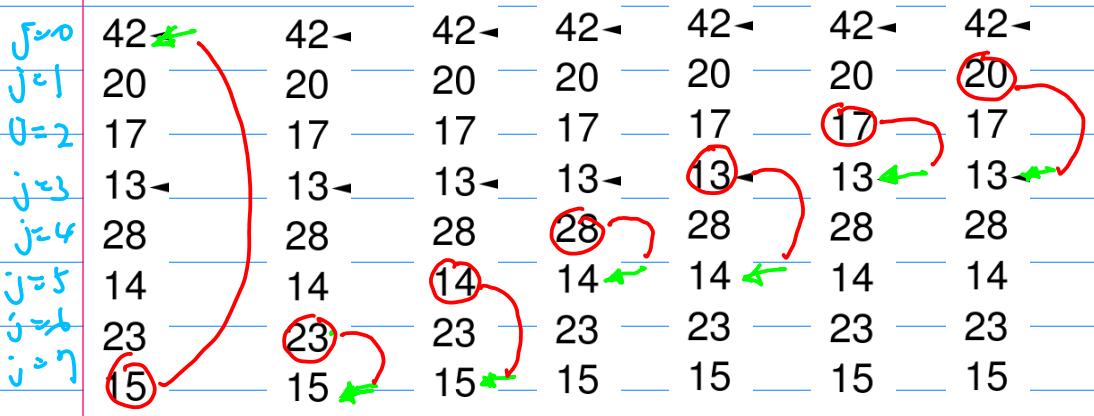
```
template <typename E, typename Comp>
void selsort(E A[], int n) { // Selection Sort
  for (int i=0; i<n-1; i++) {    // Select i'th record
    int lowindex = i;            // Remember its index
    for (int j=n-1; j>i; j--)    // Find the least value
      if (Comp::prior(A[j], A[lowindex]))
        lowindex = j;            // Put it in place
    swap(A, i, lowindex);
  }
}
```



	i=0	1	2	3	4	5	6
42	13	13	13	13	13	13	13
20	20	14	14	14	14	14	14
17	17	17	15	15	15	15	15
13	42	42	42	17	17	17	17
28	28	28	28	28	20	20	20
14	14	20	20	20	28	23	23
23	23	23	23	23	23	28	28
15	15	15	17	42	42	42	42



i=0




```
i=0 j=7
i=0 j=6
i=0 j=5
i=0 j=4
i=0 j=3
i=0 j=2
i=0 j=1
------------
i=1 j=7
i=1 j=6
i=1 j=5
i=1 j=4
i=1 j=3
i=1 j=2
------------
i=2 j=7
i=2 j=6
i=2 j=5
i=2 j=4
i=2 j=3
```

References

[1] http://en.wikipedia.org/
[2] http://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf