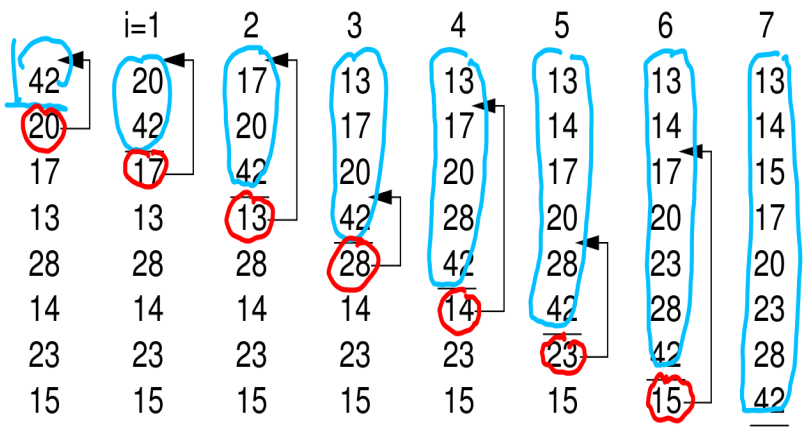
Insertion Sort

20170411

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



42

20

17

13

28

14

23

15

1-A

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

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

2 - A

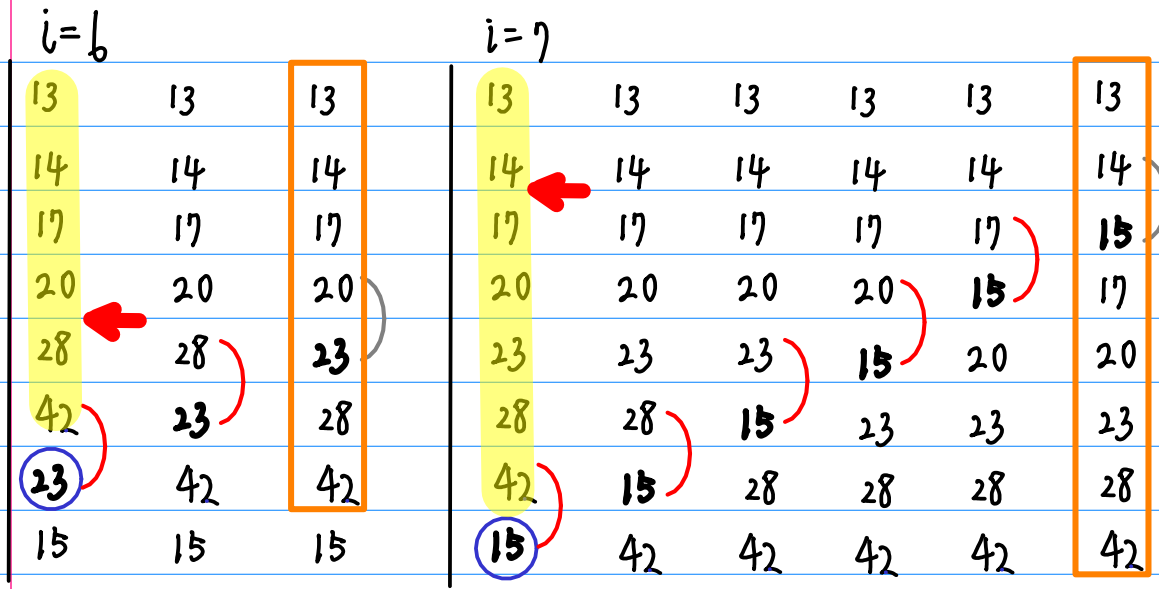
	i=6			i=7					
0	13	13	13	13	13	13	13	13	13
1	14	14	14	14	14	14	14	14	14
2	17	17	17	17	17	17	17	17	**15**
3	20	20	20	20	20	20	20	**15**	17
4	28	28	**23**	23	23	23	**15**	20	20
5	42	**23**	28	28	28	**15**	23	23	23
6	**23**	42	42	42	**15**	28	28	28	28
7	15	15	15	**15**	42	42	42	42	42

① - B

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

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

(2) - B


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

C++ template

```c
#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<float>(f);
  d2 = square<double>(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;

}
```

int
float
double

<float>
<double>

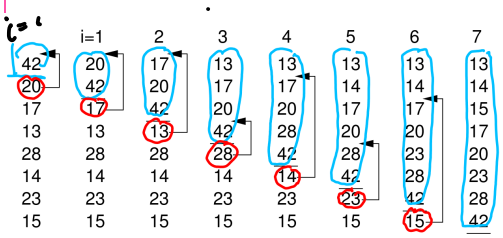
```c
template <class T>
T square(T x) {
   return (x*x);
}
```

```c
template <typename T>
T square(T x) {
  return (x*x);
}
```

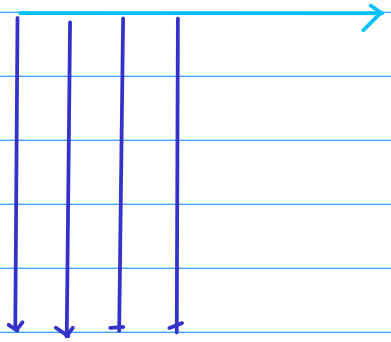
```c
square<float>(f);

int square(float x) {
   return (x*x);
}

square<int>(i);

int square(int x) {
  return (x*x);
}

square<double>(d);

double square(double x) {
   return (x*x);
}
```

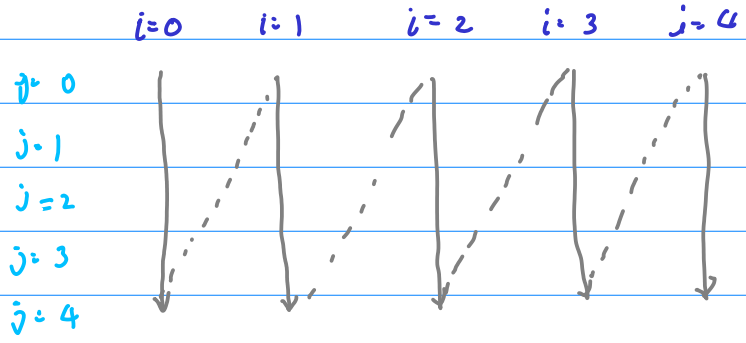
Swap

```
// 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
```

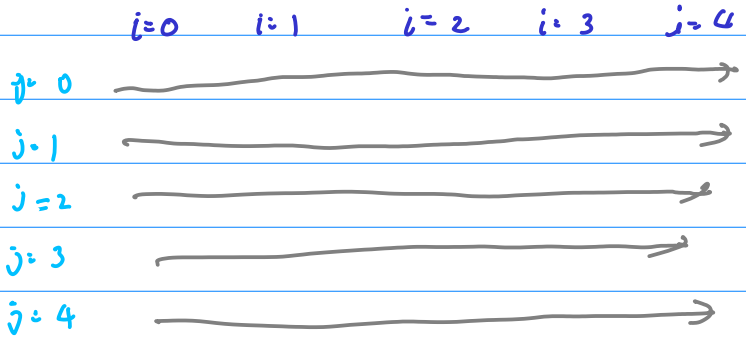



```
template <typename E, typename Comp>
void inssort(E A[], int n) { // Insertion Sort
  for (int i=1; i<n; i++)        // Insert i'th record
    for (int j=i; (j>0) && (Comp::prior(A[j], A[j-1])); j--)
      swap(A, j, j-1);
}
```

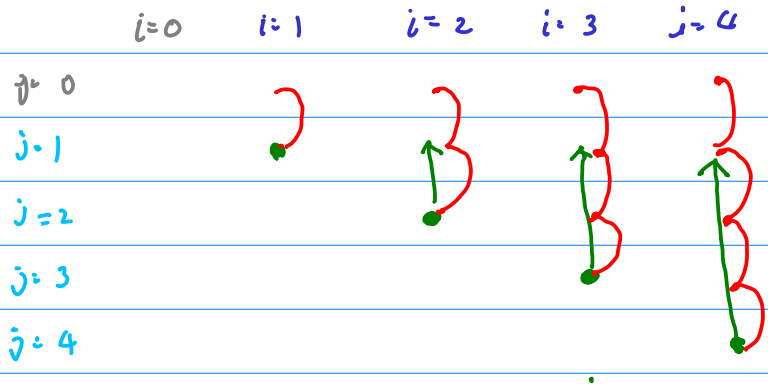




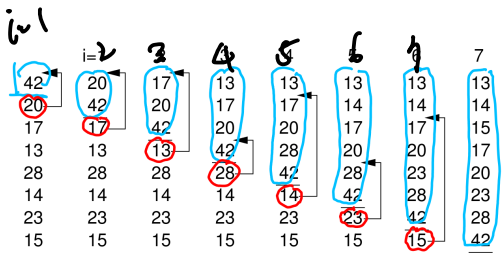

```
for (i=0; i<5; ++i)
  for (j=0; j<5; ++j)
```




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




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




```
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] ;
        }
```

(3)

$(N-1) - 2 + 1$

$(N-2) \times 2$ ← 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.

References

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