Selection Sort

Implementation:

```
void selection sort (int arr[], int n)
{
  for (int i = 0; i<(n-1); i++)
  {
    for (int j = i+1; j<n; j++)
    {
      if (arr[i] > arr[j])
        swap (arr[i], arr[j]);
    }
  }
  return;
```

Analysis:

	Cost	time
`for (int i=0; i<(n-1); i++)`	c_1	$(n-1)$
`for (int j = i+1; j<n; j++)`	c_2	$\frac{n(n+1)}{2}$
`if (arr[i] > arr[j])`	c_3	$\frac{n(n+1)}{2}$
`swap(arr[i], arr[j]);`	c_4	$\frac{n(n+1)}{2}$

```
  }
}
```

191-15-12485

∴ Time function $f(t) = c_1(n-1) + c_2\frac{n(n+1)}{2} + c_3\frac{n(n+1)}{2} + c_4\frac{n(n+1)}{2}$

Best case occurs when the array is already sorted

1	2	3	4	5

value of c_4 will be 0, so time function will be.

$f(t) = \left(\frac{c_2+c_3}{2}\right) * n^2 + \left(\frac{c_2+c_3+2c_1}{2}\right) * n - c_1$

This function look like $f(t) = an^2 + bn + c$,
which is quadric function

∴ Worst case will be occured when the array is reversely sorted

5	4	3	2	1

$f(t) = \left(\frac{c_4+c_2+c_3}{2}\right) * n^2 \left(\frac{c_4+c_2+c_3+2c_1}{2}\right) * n - c_1$

look like $f(t) = (an^2 + bn + c)$; which is quadric function.

Time complexity:

$f(t) = an^2 + bn + c$ [Best case]

$\Theta(n^2)$

Worst case:

$f(t) = an^2 + bn + c$

$\Omega(n^2)$

Insertion sort

Implementation:

```
void insertion (int ara [], int n)
{
  int i, j, x;
  for (i=0, i<n; i++)
  {
    x = ara [i],
    j = i-1;
    while (j>=0 && ara [j] >x)
      ara [j+1] =ara [j], j--;
      ara [j+1] =x;
  }
}
```

Analysis:

```
void insertion (int ara[], int n)        cost    time
{
  int i, j, x;                           c1      n
  for (i=0); i<n; i++)                   c2      n-1
  {                                      c3      n-1
    x = ara [i];                         c4      n(n+1)/2
    j = i-1;                             c5      n(n+1)/2
    while (j>0 && ara [j]>x)
    ara [j-1] = ara [j], j--;
```

```
arr[j+1] = x;
    }
}
```

∴ Time function:

$f(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \frac{n(n+1)}{2} + c_5 \frac{n(n+1)}{2} + c_6(n-1)$

Best case occurs when array is already sorted

1	2	3	5	7

when c_4 will execute for $(n-1)$ time and c_5 will execute for 0 time.

so function will be

$f(n) = c_1 * n + c_2(n-1) + c_3(n-1) + c_4(n-1) + c_5 * 0 + c_6(n-1)$

$= n(c_1 + c_2 + c_3 + c_4 + c_6) - (c_2 + c_3 + c_4 + c_6)$

∴ This look like $y = an - b$. That's Linear eqn worst case occurs when the array is reversly sorted,

7	5	3	2	1

∴ Time function will be

$f(n) = c_1 * n + c_2 + c_3(n-1) + c_4 * \frac{n(n+1)}{2} + c_5 * \frac{n(n+1)}{2} + c_6(n+1)$

$= \frac{c_1 + c_2}{2} n^2 + \frac{2(c_1 + c_2 + c_3 + c_6)}{2} n - (c_2 + c_3 + c_4 + c_6$

∴ look like $y = an^2+bn+c$; This is a quadratic eqn

Time Complexity :

Best Case:

We know, $y = an - b$ [∵ Best case]

$\Omega = \Omega(n)$

Worst Case:

We know, $y = an^2+bn+c$ [∵ Worst case]

So, $O(n^2)$