Sub.: 191-15-12195

SAT SUN MON TUE WED THU FRI

DATE: / /

Selection Sort

Implementation:

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

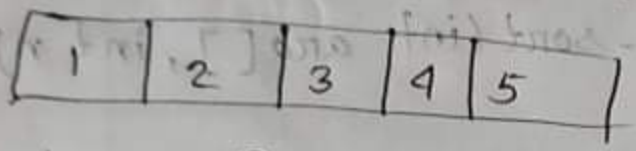
Analysis:

	cost	time
for (int i=0; i<(n-1); i++).	c1	(n-1)
{		
for (int j= i+1; j<n; j++)	c2	$\frac{n(n+1)}{2}$
{		
if (arr[i] > arr [j])	c3	$\frac{n(n+1)}{2}$
swap (arr[i], arr[j];	c4	$\frac{n(n+1)}{2}$
}		
}		

Sub.: 191-15-12145 SAT SUN MON TUE WED THU FRI DATE: / /

∴ Time function $f(t) = c1(n-1) + c2\frac{n(n+1)}{2} + c3\frac{n(n+1)}{2} + c4\frac{n(n+1)}{2}$

Best case occurs when the array is already sorted

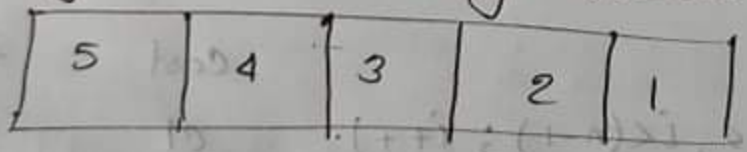


value of c4 will be 0, so time function will be

$$f(t) = \left(\frac{c2+c3}{2}\right) * n^2 + \left(\frac{c2+c3+2c1}{2}\right) * n - c1$$

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



$$f(t) = \left(\frac{c4+c2+c3}{2}\right) * n^2 + \left(\frac{c4+c2+c3+2c1}{2}\right) * n - c1$$

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

Time complexity:

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

$O(n^2)$

Sub.: 191-15-12145

SAT SUN MON TUE WED THU FRI

DATE: / /

Worst case:

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

$\Omega(n^2)$

Sub.: 191-15-12145 SAT SUN MON TUE WED THU FRI

DATE: / /

Insertion sort

Implementation:

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

Analysis:

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

```
    arr[j+1] = x;                    c6          n-1
  }
}
```

∴ Time function;

$$f(t) = c1\,n + c2\,(n-1) + c3\,(n-1) + c4\,\frac{n(n+1)}{2} + c5\,\frac{n(n+1)}{2} + c6\,(n-1)$$

Best case occurs when array is already sorted

1	2	3	5	7

when c4 will execute for (n-1) time and c5 will execute for 0 time.

so function will be,

$$f(t) = c1*n + c2\,(n-1) + c3\,(n-1) + c4\,(n-1) + c5*0 + c6\,(n-1)$$

$$= n(c1+c2+c3+c4+c6) - (c2+c3+c4+c6)$$

∴ This look like $y = an - b$, That's linear eqn

worst case occurs when the array is reversely sorted,

7	5	3	2	1

∴ Time function will be

$$f(t) = c1n + c2(n-1) + c3(n-1) + c4*\frac{n(n+1)}{2} + c5*\frac{n(n+1)}{2} + c6(n-1)$$

$$= \frac{c1+c2}{2} n^2 + \frac{2(c1+c2+c3+c6)}{2} n - (c2+c3+c$$

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

Time complexity:

Best case:

we know, $y = an - b$ [∵ Best case]

So, $\Omega(n)$

Worst case:

we know, $y = an^2 + bn + c$ [∵ worst case]

So, $O(n^2)$