ID: 191-15-12123

Selection sort

Implementation:—

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

Analysis:

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

cost time

c_1 $(n-1)$

c_2 $\frac{n(n+1)}{2}$

c_3 $\frac{n(n+1)}{2}$

c_4 $\frac{n(n+1)}{2}$

191-15-12123

∴ 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 $f(t) = ax^2+bx+c$

which is quadric function.

∴ Worst case will be occured when the array is reversly 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 $f(t) = (ax^2+bx+c)$; which is quadric function.

Time complexity:

$f(t) = ax^2+bx+c$ [Best case]

$O(n^2)$

Worst case:

$f(t) = ax^2+bx+c$

$\Omega(n^2)$

191-15-12123

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

arr[j+1] = x; c_6 $n-1$

}

}

∴ Time function:

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

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 $y = ax - b$, That's linear eqn worst case occurs when the array is reverse sorted.

7	5	3	2	1

∴ Time function will be

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

$c_6(n-1)$

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

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

Time complexity:

Best case

we know, $y = ax - b$ [∵ Best case]

So, $\Omega(n)$

Worst case:

we know, $y = ax^2 + bx + c$ [∵ worst case]

So, $O(n^2)$.