cpts 350

HW 3.

1.)

```
Partition (A, p, q).
{
        a = A[p]
        i = p.
        for (j = p+1; j ≤ q; j++).{
            if( A[j] ≤ a){
                i++;
                swap A[i] A[j]
            }
        swap A[i] A[j]
        return i;
        }
}
```

2.) the input array has 1% probability to be monotonically decreasing:

when the array is recursive sorted

$$T_{Avg}(n) = 1\% \ \Theta(n^2) + 99\% \ O(n^2)$$

to show: $t_{Avg} = \Theta(n^2)$ ie for some $c, c' > 0$

$$cn^2 \leq T_{Avg} \leq c'n^2.$$

Step 1: $t_{Avg}(n) \leq 1\% \cdot cn^2 + 99\% \ cn^2$

$\leq cn^2$ for some $c$.

hence upperbound.

Step-2:- $t_{Avg}(n) \geq 1\% \ \Theta(n^2)$.

$\geq 1\% \ cn^2$ for some $c$

3.) the partition algorithm will ~~transf~~ traverse through each element of the array through recursion.

hence, the best, average, worst case time complexity will $O(n)$

quicksort : (for n elements).

Best case :- $O(n \log n)$

worst case :- $O(n^2)$.

Average case :- $O(n \log n)$.

Insert Sort :- (for n-> elements).

Best case :- $O(n)$

worst case :- $O(n^2)$

Average case : $O(n^2)$.

Average case:

$$T_{Avg} = \frac{1}{n} \sum_{r=1}^{n} \left\{ O(n-r)^2 + O(r-1)\log(r-1) \right\} + O(n)$$

now we need to find a constant $c$

such that $T_{AVG} \leq cn^2$

$$T_{AVG} = \frac{1}{n} \sum_{r=1}^{n} \left\{ a(r-1)\cdot\log(r-1) + a(n-r)^2 \right\} + an$$

$$= \frac{a}{n} \sum_{r=1}^{n} \left\{ (r-1)\log(r-1) \right\} + \frac{a}{n} \sum_{r=1}^{n} (n-r)^2 + an.$$

$$= \frac{2a}{n} \sum_{r=1}^{n} (r-1)^2 + an$$

$$\leq \frac{2a}{n} \frac{1}{3} n^3 + an = O(n^2).$$

worst case:

$$T_w = \max_{1 \le r \le n} \{ O(n-r)^2 + O(r-1)^2 + O(n)$$

$$\max_{1 \le r \le n} \{ a(n-r)^2 + a(r-1)^2 \} + an$$

$$\max_{1 \le r \le n} \{ \underbrace{an^2 + ar^2 - 2anr + a(r^2 + 1 - 2r)}_{F(r)} \} + an.$$

$$\therefore F(r) = an^2 + ar^2 - 2anr + a[r^2 + 1 - 2r]$$

$$F'(r) = 0 + 2ar - 2an + a[2r - 2]$$
$$= 2ar - 2an + 2ar - 2.$$

$$F''(r) = 2a + 2a = 4a > 0$$

hence

$\therefore$ we will consider max case:

$$\Rightarrow \max [F(1), F(n), F(n)] + an.$$

$$F(1) + an$$
$$\Rightarrow a(n-1)^2 + an$$
$$= O(n^2)$$

<u>Best case:</u> the best case will be consider when the partion index as in the middle.

$$T_B = O(n) + O(n/2) + O\left(\frac{n}{2} \log \frac{n}{2}\right).$$

$$= an + a\frac{n}{2} + a\frac{n}{2} \log \frac{n}{2}.$$

$$= \frac{3an}{2} + a\frac{n}{2} \log \frac{n}{2}.$$

hence, it will be $\Theta(n)$

4.) Best case: for the ~~the~~ best case, the array should be recursive sorted or random.

hence, the best case time complexity will be $\Theta(n)$ since the array will be transversed once.

worst case:

$$T_w(n) = \max_{1 \le x \le n} \{ T_w(x-1) + O(n-x)^2 \} + O(n)$$

hence, we need to find a constant $c$ such that $T_w(n) \le cn^2$

check I.H. $\forall\ i < n$, $T_w(n) \le ci^2$.

$$T_w(n) = \max_{1 \le x \le n} \{ T_w(x-1) + O(n-x)^2 \} + O(n)$$

$$\le \max_{1 \le x \le n} \{ \underbrace{c(x-1)^2 + a(n-x)^2 + an}_{F(x)} \}$$

$$F'(x) = 2c(x-1) - 2a(n-x)$$

$$F''(x) = 2c + 2a > 0 \qquad F(x) \text{ is } \cup$$

hence we will consider only end cases

$$\Rightarrow \max \{ F(1), F(n) \} + a \cdot n.$$

$$\Rightarrow \max \{ a(n-1)^2, c(n-1)^2 \} + a \cdot n.$$

$$\Rightarrow c(n-1)^2 + a \cdot n$$

$cn^2$ when $c > a$.

Average case:

$$T_{Avg}(n) = \frac{1}{n}\sum_{r=1}^{n}\left\{T_{Avg}(r-1) + O(n-r)^2 + O(n)\right\}.$$

$$T_{Avg}(n) = O(n^2) \le Cn^2 \text{ for some } C>0.$$

check I.H $\forall\, i<n$, $T_{Avg}(n) \le Ci^2$.

$$\le \frac{1}{n}\sum_{r=1}^{n}\left\{c(r-1)^2 + a(n-r)^2\right\} + a\cdot n.$$

$$= \frac{1}{n}\sum_{r=1}^{n} c(r-1)^2 + \frac{1}{n}\sum_{a.1}^{n} a(n-r)^2 + \frac{1}{n}\sum_{r=1}^{n} an.$$

$$\Rightarrow \frac{c}{n}\sum_{r=1}^{n}(r-1)^2 + \frac{a}{n}\sum_{r=1}^{n}(r-1)^2 + a\cdot n$$

$$\Rightarrow \frac{a+c}{n}\sum_{r=1}^{n}(r-1)^2 + a\cdot n$$

$$\le \frac{a+c}{n}\int_0^n x^2\,dx + a\cdot n.$$

$$= \frac{a+c}{n}\cdot\frac{1}{3}n^2 + an.$$

$$= \frac{a+c}{3}n^2 + a\cdot n$$

$$\le Cn^2 \quad \text{when } C \gg a.$$

$\Rightarrow O(n^2)$ is proved.