Neslihan Çekiç
150160060

ANALYSIS OF ALGORITHM I

HOMEWORK 1

Problem: Let $A[1..n]$ be an array of n numbers. If $i<j$ and $A[i]=A[j]$, then the pair (i,j) is called a duplication of A.

Part (a):

To compute the expected number of duplicants, we must use indicater random variables. Let's call $X_{i,j}$ our indicater random variable for this problem as inc the equation below.

for $1 \le i < j \le n \Rightarrow X_{i,j} = I\{A[i] = A[j]\}$

The probability of $A[i]=A[j]$ is would be $\frac{1}{n}$ for this problem, because in a case where $n=50$, let's say, our first number came 15. The probability that the second number will be 15 again would be $\frac{1}{50}$.

$$E[X_{i,j}] = Pr\{A[i] = A[j]\} = \frac{1}{n}$$

Let's called X, the random variable in the array, indicating the total number of $X_{i,j}$'s.

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{i,j}$$

In this case $\underline{E[X]}$ gives us the expected number of duplicats.

$$E[X] = E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{i,j}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{i,j}]$$

$$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{n} \longrightarrow \sum_{j=i+1}^{n} \frac{1}{n} = \frac{n-(i+1)+1}{n} = \frac{n-i}{n}$$

$$E[X] = \sum_{i=1}^{n-1} \frac{n-i}{n} = \frac{1}{n} \sum_{i=1}^{n-1} n-i \longrightarrow \sum_{i=1}^{n-1} n = n.(n-1)$$

$$\longrightarrow \sum_{i=1}^{n-1} i = \frac{(n-1).n}{2}$$

$$E[X] = \frac{1}{n}\left(n.(n-1) - \frac{n.(n-1)}{2}\right)$$

$$E[X] = \frac{1}{n} \cdot \frac{n.(n-1)}{2}$$

$$E[X] = \frac{n-1}{2} \quad \rightarrow \text{Expected number of duplicants.}$$

Neslihan Çekiç
150160060

Part (b):

```
int duplicate(int** a, int n){
C1   int count=0;                                   1
C2   for(int i=1; i<=n-1; i++){                     Σ_{i=1}^{n-1}
C3       for(int j=i+1; j<=n; j++){                 Σ_{i=1}^{n-1} Σ_{j=i+1}^{n}
C4           if(a[i][0]==a[j][0]){
C5               cout << '(' << i << ',' << j << ')' << "/t";
C6               count++;
C7               if(count%10==0){
C8                   cout << endl;
C9               }
             }
         }
     }
C10  return count;
}
```

in A[i]=A[j] control statements the lines depends on input so we calculate running time for 3 case.

Worst case T(n)

→ in worst case all numbers are the same so every A[i]=A[j] is true

	cost	no. of times
C1	1	1
C2	2	$\sum_{i=1}^{n-1} \to n$
C3	2	$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \to \frac{n.(n-1)}{2}$
C4	1	$\frac{n.(n-1)}{2}$
C5	1	"
C6	1	"
C7	1	"
C8	1	"
C9	1	$\frac{n.(n-1)}{20}$
C10	1	1

$$T(n) = 1+2n+n(n-1)+\frac{5.n(n-1)}{2}+\frac{n.(n-1)}{20}+1$$

$$T(n) = \frac{71n^2}{20}-\frac{31n}{20}+2 \quad \text{(Worst case)}$$

Best Case T(n)

→ in best case there is no duplicants so in control point doesn't executed at all.

	cost	no of times
C1	1	1
C2	2	n
C3	2	$\frac{n(n-1)}{2}$
C4	1	$\frac{n.(n-1)}{2}$
C5	0	-
C6	0	-
C7	0	-
C8	0	-
C9	0	-
C10	1	1

$$T(n) = 1+2n+n.(n-1)+\frac{n.(n-1)}{2}+1$$

$$T(n) = \frac{3n^2}{2}+\frac{n}{2}+2 \quad \text{(best case)}$$

Avarage Case T(n)

→ In average case we use expected number of duplicants for average T(n) which we found it in previous part.

C_1	1	1
C_2	2	n
C_3	2	$\frac{n.(n-1)}{2}$
C_4	1	$\frac{n.(n-1)}{2}$
C_5	1	$\frac{n-1}{2}$
C_6	1	"
C_7	1	"
C_8	1	"
C_9	1	$\frac{n-1}{20}$
C_{10}	1	1

$$T(n) = 1+2n+n.(n-1)+\frac{n.(n-1)}{2}+\frac{4.(n-1)}{2}+\frac{n-1}{20}+1$$

$$T(n) = \frac{3}{2}n^2+\frac{51n}{20}-\frac{1}{20} \quad \text{(avarage case)}$$

* As we see in running times the complexity of this algorithm is $n^2 \Rightarrow O(n^2)$