Assignment

Course code: CSE-214/215

Course Title: Algorithms

Submitted To

Subroto Nag Pinku

Department of CSE

Daffodil International University

Submitted by

Name: Mahmodul Hasan

ID: 191-15-12933

Section: O-14

Department of CSE

Daffodil International University

Date of submission: 08.11.2020

1.Write short notes on Optimization.

Answer: In computing, optimization is the process of modifying a system to make some features of it work more efficiently or use fewer resources. For instance, a computer program may be optimized so that it runs faster, or to run with less memory requirements or other resources or to consume less energy. This is a branch of software engineering.

2.What are the different algorithms you know.

Answer: I know some algorithms. Such as Binary search, Linear search, Quick sort, Merge Sort and etc.

a. Binary Search Algorithm: Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeatedly check until the value is found or the interval is empty.

b. Linear search Algorithm: Start from the leftmost element of arr[] and one by one compare x with each element of arr[].If x matches with an element, return the index. If x doesn't match with any of elements, return -1 (Not Found).

c. Why are you learning so many algorithms?

We are learning so many algorithms to use those basic methods to give more efficient solution for codes or complex problems we do everyday.

Algorithms are the solution steps for a particular problem. Writing algorithm helps me to boost up my thinking ability for other programing problems. Also algorithm help me to pradict the program will work or not.

d. Show analysis of a recursive algorithm.

```
int Factorial (int n) {
if (n == 0)
  return 1;
```

```
else
  return n*Factorial(n-1);
}
```

Time complixity analysis:

T(n)= T(n-1)+1+1+1

=T(n-1)+3

=T(n-2)+6

=T(n-3)+8

=T(n-k)+3k

Now,

n-k=0

n=k

T(n)=T(0)+3n [n=k]

T(n)=3n+1

The Time complexity in the worst case: O(n)

e. Design an iterative and recursive algorithm and prove that your algorithm works.

Recursive Fibonacci vs Iterative method

```
Int fib int(n)
{
If(n<=1)
Return n;
Return fibo (n-1) + fibo(n-2);
}
```

Iterative method:

```
Int fib(int n)
{
Int f(n+2);
Int i;
f[0]=0;
f[1]=1;
for(i=2; i<=n; i++)
{
F[i]= f[i-1] + f[i-2];
}
Return f[n];
}
};
```

Time Complexity analysis:

T(n)= T(n-1)+ T(n-2)+ 1

We can assume that T(n-2)= T(n-1).

Substituting the value of T(n-1) = T(n-2) into our relation T(n), we get:

T(n)= T(n-1)+ T(n-1)+1= 2*T(n-1)+1

T(n)= 2*[2*T(n-2)+1]+1= 4*T(n-2)+3

Next, we can substitute in T(n-2)= 2*T(n-3)+1:

T(n)= 2*[2*[2*T(n-3)+1]+1]+1]=8*T(n-3)+7

And again for T(n-3)= 2*T(n-4)+ 1:

T(n)= 2*[2*[2*[2*T(n-4)+1]+1]+1]+1= 16*T(n-4)+ 15

We can see a patten starting to emerge here, so let's attempt to form a general solution forT(n). It appers to stand that:

T(n)= 2^k*T(n-k)+ (2^k-10

In this function 2^k is the term that has the highest rate of change for differentt values. That's why we can assume that

Time complexity is (2^n)

While the iterative method is O(n).