Algorithms and Complexity COMP\_90038

Semester 2 August2020

Name:Sakshi Chandel

Student ID:1124298

Question 1:

Solution 1:

**Solution 1(a):**

The basic idea of this algorithm is to initialize an array Z of size such that it can have elements of both array X and Y(size of array X+ size of array Y).Then from array X put all elements to array Z .

Then check each element of array Y if it is present in array X using binary search .If present ,don’t add to array Z else add to array Z. Hence will get union of elements from array X and array Y in array Z.

procedure FindSetUnion(X,Y )

n=size of array X

initialize array Z with size 2n;

for i ← 0 to i ←n-1 do :

Z[i]  ← X[i];

for j  ← 0 to j  ← n-1 do :

index  ← BinarySearch(X,0,m,Y[j]); #complexity of binary search -> logn

if(index=-1) #element Y[j] is not in array X

Z[k]  ← Y[j];

k++;

**Since here size of both arrays is n that is why complexity will be n+nlogn,**

**which is in O(nlogn).**

# **Question 1:**

**Solution 1 (b):**

procedure FindSetUnion(X,Y)

initialise hashtable as hash\_table

n ← size of X #size of X=size of Y =n

for i ← 0 to i ← n-1 do :

insert(hash\_table,X[i])

for j ← 0 to j←n-1 do:

result ← search(hash\_table,Y[j])

if(result=false)

insert(hash\_table,Y[j])

Overall complexity = O(n)

Question 3:

Solution 3

1. **Recurrence relation:** T(N) = 1 + Sum j = 1 to N-1 (T(j))

procedure lis(A, n)

if (n == 1) then return 1;

res, max\_ending\_here = 1;

for (int i = 1; i < n; i++)

res = lis(arr, i);

if (arr[i-1] < arr[n-1] && res + 1 > max\_ending\_here)

max\_ending\_here = res + 1;

if (max\_ref < max\_ending\_here)

max\_ref = max\_ending\_here;

return max\_ending\_here;

procedure LongestIncreasingLengths (arr[],curr)

if(curr == 0) then return 1; // Only one subsequence ends at first index, the number itself

ans ← 1

for(i ← 0 to i ← curr-1 ) do:

if(arr[i] < arr[curr])

ans = max(ans, 1 + LongestIncreasingLengths(arr, i))

return ans

**c)Time Complexity:**O(2^N)

The time complexity is exponential. There will be 2^n - 1 nodes will be generated for a n sized array.