CSE 321 Homework - 3

Saut Nalbant
1801042614

a) Algorithm alg1(L[0...n-1])

if(n==1) return L[0] → O(1)

else

tmp = alg1(L[0...n-2]) → T(n-1)

if(tmp<=L[n-1]) return tmp → O(1)

else return L[n-1] → O(1)

$T(n) = T(\cancel{n-1}) + O(1)$

$T(\cancel{n-1}) = T(\cancel{n-2}) + O(1)$

$T(\cancel{n-2}) - T(\cancel{n-3}) + O(1)$

$\vdots$

$+ T(\cancel{1}) = T(0) + O(1)$

$\left.\right\} (n-1) \cdot O(1)$

$O(1)$

$T(n) = T(0) + (n-1)$

$T(n) = O(n-1)$

$\boxed{T(n) = O(n)}$

O(n

b) Algorithm alg2(X[l...r])

if(l==r) return X[l] → O(1)

else

flr = floor((l+r)/2) → O(1)

tmp1 = alg2(X[l...flr]) → $T(\frac{n}{2})$

tmp2 = alg2[X[flr+1..r) → $T(\frac{n}{2})$

if(tmp1 <= tmp2) return tmp1 → O(1)

else return tmp2 → O(1)

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

according to master theorem $a = 2$ $b = 2$ $f(n) = O(1)$

$n^{\log_2 2} = n$ $f(n) \in O(n^{\log_b a}) \Rightarrow \Theta(n)$

2) You are a given a polynomal $p(x)$ like

$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots a_1 x + a_0$ you are supposed to write a brute force algorithm for computing the value of the polynomal at given point x_0.

```
algorithm ( P[0... n], n )
    result = 0
    for i=n to 0
        mult = 1
        for j=1 to n
            mult = mult * n          O(1)
        result = result + P[i] * mult   O(1)
    return result
```

$$\left.\begin{array}{l}\text{mult} = \text{mult} * n \quad O(1) \\ \text{result} = \text{result} + P[i] * \text{mult} \quad O(1)\end{array}\right\} \Theta(n-1)$$

$$\text{for loop} \Big\} \Theta(n)$$

$$T(n) = \sum_{i=n}^{0} \sum_{j=1}^{n} \Theta(1) = ((n-1)+1) \cdot \Theta(1) \quad \sum_{i=n}^{0} \Theta(n) = n \cdot \Theta(n)$$

$$\boxed{T(n) = \Theta(n^2)}$$

CamScanner ile tarandı

algorithm (points[0...n])
min dist = inf
for i=0 to n → $\Theta(n)$
for j = i+1 to n
if (get Distance (points [i], points [j]) $\Theta(1)$
min = get Distance (points [i], points[j]) $\Theta(1)$ } $\Theta(n)$
return min

get Distance = $\Theta(1)$ algorithm
assume points has two field
x and y

$$T(n) = \sum_{i=0}^{n} \sum_{i+1}^{n} \Theta(1) = (n-(i+1)+1)\Theta(1) = \Theta(n)$$

$$\sum_{i=0}^{n} \Theta(n) = \Theta(n) \cdot n = \Theta(n^2)$$

5) Algorithm(stations [0... n])
list [] most_profit , consecutive;
for i to n
clear (consecutive;
if stations [i]. profit > total-profit(most-profit)
clear most - profit
most-profit. add (station[i])
consecutive .add(station [i])
for j=i+1 to n
consecutive . add(station [i])
if(total-profit(consecutive)> total_profit(most-profit)
swap (total-profit, most-profit)

return most -profit

$$T(n) = \sum_{i=0}^{n} \sum_{j=i+1}^{n} \underbrace{2\Theta(n)+\Theta(n)}_{\Theta(n)}$$

$$(n-(i+1)+1)\,\Theta(n)$$

swap and total-profits are $\Theta(n)$ algorithms.

$$T(n) = \Theta(n^3) \Leftarrow (n+1)\cdot\Theta(n)(n-i) \Leftarrow \sum_{i=0}^{n} \Theta(n)(n-i)$$