Innopolis University, S19

Assignment 2. Theory part

Data Structures and Algorithms

Manakovskiy
Daniil.
BS18-02

1) Applications of Master Theorem

a) $T(n) = 16\,T(\frac{n}{4}) + n$

$a = 16,\ b = 4,\ f(n) = n$

$f(n) = O(n^{\log_b a - \varepsilon}) = O(n^{\log_4 16 - \varepsilon}) = O(n^{2-\varepsilon})$ for $\varepsilon = 1$

Then $T(n) = \Theta(n^2)$

b) $T(n) = T(\frac{n}{2}) + 2^n$

$a = 1,\ b = 2,\ f(n) = 2^n$

$n^{\log_b a} = n^{\log_2 1} = n^0 = 1$

$f(n) = \Theta(2^n) = \Omega(n^{0+\varepsilon})$ for $\varepsilon = 1$

$f(\frac{n}{2}) \le c f(n)$

$2^{n/2} \le c\, 2^n$

$\frac{1}{2^{n/2}} \le c$

since $n > 0$, $\frac{1}{2^{n/2}} \in (0; 1) \rightarrow \frac{1}{2^{n/2}} < 1 \Rightarrow \exists c < 1$ that satisfies inequality

$\Downarrow$

$T(n) = \Theta(f(n)) = \Theta(2^n)$

c) $T(n) = 2T(\frac{n}{2}) + \frac{n}{\log n}$

$n^{\log_b a} = n^{\log_2 2} = n$

$\frac{n/\log n}{n} = \frac{1}{\log n} \neq n^c\ \forall c \in \mathbb{R} \Rightarrow$

$\Rightarrow$ there is non-polynomial difference between $f(n)$ and $n^{\log_b a}$ => The master theorem is not applicable

d) $T(n) = 4T\left(\frac{n}{2}\right) + n^2$

$a = 4$ $b = 2$ $f(n) = n^2$

$n^{\log_b a} = n^2$

$f(n) = n^2 = \Theta(n^2) \Rightarrow T(n) = \Theta(n^2 \log n)$

2) Solving a problem (dynamic prog.)

1) Let dist[i] be a minimum distance from station 1 to station i:

$1 \le i \le n$, dist[n] is the total cost

2) The step:

$$dist[i] = \begin{cases} 0, & i = 1 \\ \min(\{d[j] + f_{ji} \mid \forall j : 1 \le j < i\}), & i \ne 1 \end{cases}$$

3) The proof of correctness:

Strong induction:

a) base case: dist[1]=0 – true, because you do not have to move from station 1 in order to arrive to station 1

b) Suppose that dist[1], dist[2], ..., dist[k] are minimum distances from station 1 to stations 1, 2, ..., K correspondingly

c) Prove that $dist[K+1] = \min(\{dist[i] + f_{i,K+1} \mid \forall i: 1 \le i \le K\})$ is the minimum distance from station 1 to K+1:

Since we supposed that dist[1]... dist[K] are lengths of shortest distances to stations 1..K, then $\min(dist[1]+f_{1,K+1}, \dots)$ is indeed the shortest path

4) As was shown in step 2 and prooved at step 3.a, the base case is calculating a distance to the source statiton. Distance is always 1, therefore, computation complexity $O(1)$

5) int distance[1.. N_STATIONS]

```
1. distance [1] = 0                                      // 1
2. for i = 2 to N_STATIONS do                            // n
3.     int min_dist = INFINITY                           // n-1
4.     for j = i-1 down to 1 do                          // Σ_{i=2}^{n} i
5.         min_dist = min(min_dist, distance[j] + f_{j,i}) // Σ_{i=2}^{n} (i-1)
6.     end
7.     distance[i] = min_dist                            // n-1
   end
```

$$T(n) = C_1 + n \cdot C_2 + (n-1)C_3 + C_4 \cdot \sum_{i=2}^{n} i + C_5 \sum_{i=2}^{n}(i-1) + C_7(n-1)$$

$$T(n) = C_1 + n \cdot C_2 + (n-1)C_3 + C_4 \frac{(n+2)(n-1)}{2} + C_5 \frac{n(n-1)}{2} + C_7(n-1)$$

After combining like terms it becomes obvious that $T(n)$ is a 2^{nd} degree polynomial $\Rightarrow T(n) = O(n^2)$