CSE 321 - Homework 5

Serika Sera TOPKAYA
131044020

(1) Since it is dynamic programming I try to find a recurrence relation first. I assumed that possibility selection into two groups. One those includes the last element and other one does not.

(a) Recurrence relation is:

$$F(n) = \max(arr(n), F(n-1) + arr(n))$$

Time Complexity:

$$T(n) = \sum_{i=1}^{n} c = n \Rightarrow \boxed{T(n) = \Theta(n)}$$

(b) In homework 3, in option (a) I designed Brute force algorithm and its time complexity was $\Theta(n^3)$. In option (b) I designed a Divide&Conquer algorithm and its time complexity was $\Theta(n \log n)$. But this time I designed Dynamic Programming and the algorithm's time complexity is $\Theta(n)$. So, dynamic programming is much better in terms of running time.

(2) Since it is dynamic programming I try to find a recurrence relation first. Then I design algorithm. I create DP table and computed all possible pairs that their sum may give the given candy length. I filled the DP table with maximum obtainable value. Then I returned maximum obtainable value of which requested candy length. (Bottom-up approach)

Recurrence relation:

$$F(n) = \max(arr(i) + F(n-i))$$

Time complexity:

$$T(n) = \sum_{i=0}^{n} \sum_{j=0}^{i} c = \sum_{i=0}^{n} i.c = \frac{n(n+1)}{2} = \frac{n^2+n}{2} \in n^2$$

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

(3) According to my research this kind of problems should be solved with fractional greedy. So after I filled the box with the cheeses with the largest cost, I continued to fill the box by dividing the cheeses into portions.

Time complexity:

$$T(n) = \sum_{i=0}^{n} c + T(sort) + \sum_{i=0}^{n} c$$

$$= T(n) + T(n \log n) + T(n)$$

↳ sorting time.

$$\boxed{T(n) = \Theta(n \log n)}$$

(4) First I assumed that the first course was selected. So, I start counting from 1. Then I check if the start time of the current course is later than the finish time of the previous course in a loop. If condition is true I increased the counter. Then I returned the counter as a result.

Time complexity:

$$T(n) = \sum_{i=0}^{n} c \rightarrow \text{constant}$$

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