Quiz 2: Solutions
CSE 551
Fall 2020

1) The optimal cost of constructing a binary search tree with 5 key values [B, F, J, P, V] with the following set of frequencies for successful searches [64,84,52,64,24] and unsuccessful searches [0,68,76,116,190,42] is?

ANS: 1872. Check next page for the solution.

ANSWER 1 :

→ Sequence of key values : [B, F, J, P, V]

→ Frequencies of successful search, β = [64, 89, 52, 64, 24]

→ Frequencies of unsuccessful search, α = [0, 68, 76, 116, 190, 42]

→ Let C_{ij} be the cost of optimal binary search tree over the respective frequencies, where :

→ $C_{ii} = 0$

→ $C_{ij} = \min_{i<k\leq j} (C_{i,k-1} + C_{k,j}) + \sum_{t=i}^{j} \alpha_t + \sum_{t=i+1}^{j} \beta_t$

→ Let R_{ij} = value of k that minimizes $C_{i,k-1} + C_{k,j}$

→ Let weights be represented by ω, where

→ $\omega_{ii} = \alpha_i$

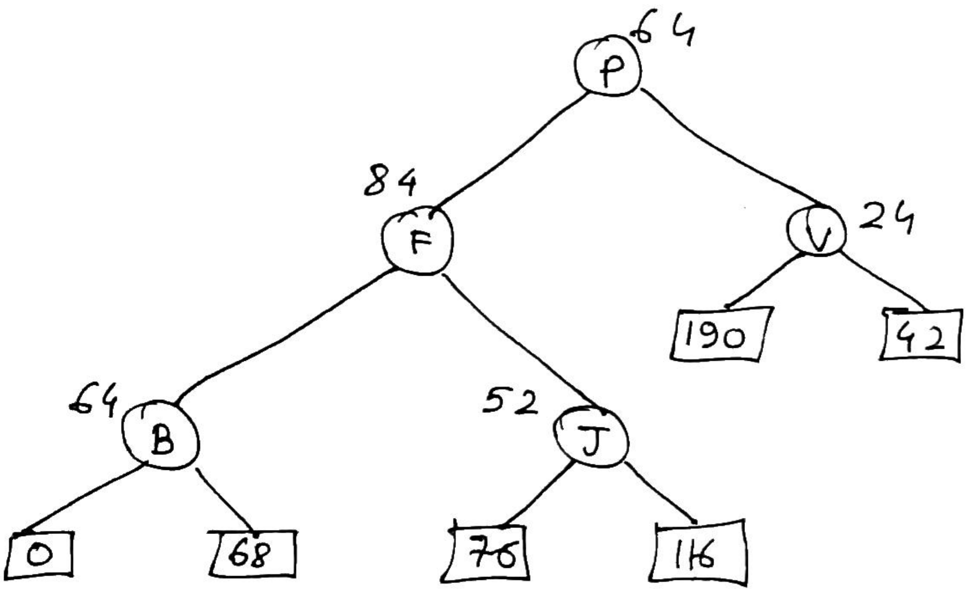
→ $\omega_{ij} = \omega_{i,j-1} + \beta_j + \alpha_j \quad , i < j$

$\therefore \; C_{ij} = \min_{i<k\leq j} ((C_{i,k-1} + C_{k,j}) + \omega_{ij}$

Now, for all values of i and j ranging from 0 to 5,
putting all the values in a table :

j \ i	0	1	2	3	4	5
5	$R_{05} = 4$ $w_{05} = 780$ $C_{05} = 1872$	$R_{15} = 4$ $w_{15} = 716$ $C_{15} = 1576$	$R_{25} = 4$ $w_{25} = 564$ $C_{25} = 1044$	$R_{35} = 4$ $w_{35} = 436$ $C_{35} = 692$	$R_{45} = 5$ $w_{45} = 256$ $C_{45} = 256$	$R_{55} = 0$ $w_{55} = 42$ $C_{55} = 0$
4	$R_{04} = 3$ $w_{04} = 714$ $C_{04} = 1508$	$R_{14} = 3$ $w_{14} = 650$ $C_{14} = 1248$	$R_{24} = 4$ $w_{24} = 498$ $C_{24} = 742$	$R_{34} = 4$ $w_{34} = 370$ $C_{34} = 370$	$R_{44} = 0$ $w_{44} = 190$ $C_{44} = 0$	
3	$R_{03} = 2$ $w_{03} = 460$ $C_{03} = 836$	$R_{13} = 3$ $w_{13} = 396$ $C_{13} = 624$	$R_{23} = 3$ $w_{23} = 244$ $C_{23} = 244$	$R_{33} = 0$ $w_{33} = 116$ $C_{33} = 0$		
2	$R_{02} = 2$ $w_{02} = 292$ $C_{02} = 424$	$R_{12} = 2$ $w_{12} = 228$ $C_{12} = 228$	$R_{22} = 0$ $w_{22} = 76$ $C_{22} = 0$			
1	$R_{01} = 1$ $w_{01} = 132$ $C_{01} = 132$	$R_{11} = 0$ $w_{11} = 68$ $C_{11} = 0$				
0	$R_{00} = 0$ $w_{00} = 0$ $C_{00} = 0$					

The required optimal binary search tree is :



2) We have a knapsack with capacity C, and a set of n objects $O = o_1, ..., o_n$. Each object oi has a size s_i and a value v_i associated with it. You are required to select a subset of objects such that the sum of the sizes of the selected objects will not exceed the knapsack capacity C, and the sum of the values of the selected objects is maximized.

State True/False:

The following strategy will yield the optimal solution:

Sort the objects by decreasing values. Select the objects in order of decreasing value, until the knapsack cannot hold any more objects.

ANS: False.

Suppose that we have a capacity of 50. We are given the following size, value pairs: (50, 70), (20, 60) and (30, 40). If we select objects by decreasing values, we will select the object with size50 and value 70. Our knapsack is now full. But the optimal is (20,60) + (30, 40) for a value of 100.

3) We have a knapsack with capacity C, and a set of n objects $O = o_1, ..., o_n$. Each object oi has a size s_i and a value v_i associated with it. You are required to select a subset of objects such that the sum of the sizes of the selected objects will not exceed the knapsack capacity C, and the sum of the values of the selected objects is maximized.

State True/False:

The following strategy will yield the optimal solution:

Sort the objects by increasing sizes. Select the object in order of increasing size, until the knapsack cannot hold any more objects.

ANS: False.

Suppose that we have a capacity of 50. We are given the fol-lowing size, value pairs: (50, 60), (20, 10) and (30, 20). If we select objects by increasing sizes, we will select the object with sizes 20 and value 10 and 30 with value 20. Our knapsack is now full. But the optimal is (50, 60) for a value of 60.

4) We have a knapsack with capacity C, and a set of n objects $O = o_1, ..., o_n$. Each object oi has a size s_i and a value v_i associated with it. You are required to select a subset of objects such that the sum of the sizes of the selected objects will not exceed the knapsack capacity C, and the sum of the values of the selected objects is maximized.

State True/False:

The following strategy will yield the optimal solution:

Sort the objects by decreasing ratio of $\frac{v_i}{s_i}$. Select the object in order of decreasing ratio of $\frac{v_i}{s_i}$, until the knapsack cannot hold anymore objects.

ANS: False.

Suppose that we have a capacity of 50. We are given the following size, value pairs: (10, 30), (50, 100) and (50, 50). If we select objects by decreasing ration, we will select the object with size10 and value 30. But the optimal is (50, 100) for a value of 100.

5) Let $A_1, A_2, ..., A_n$ be matrices where the dimensions of A_i are $d_{i-1} * d_i$ for $i = 1, ..., n$. Here is a strategy to compute the best order (i.e., the order that requires the fewest number of multiplications) in which to perform the matrix multiplications to compute $A_1, A_2, ..., A_n$.

At each step, choose the largest remaining dimension (from among $d_1, ..., dn - 1$) and multiply two adjacent matrices that share the same dimension.

State True/False:

The above strategy will always minimize the number of multiplications necessary to multiply the matrix chain $A_1, ..., A_n$.

ANS: False.

Consider the following three matrices: M_1, M_2, M_3 with dimensions $1 \times 1, 1 \times 2, 2 \times 3$. Following the above strategy, we would end up multiplying $M_1 \times (M_2 \times M_3)$ for a total of 9 multiplications. However, if we multiply $(M_1 \times M_2) \times M_3$, we would have 8 multiplications.

6) Let $A_1, A_2, ..., A_n$ be matrices where the dimensions of A_i are $d_{i-1} * d_i$ for $i = 1, ..., n$. Here is a strategy to compute the best order (i.e., the order that requires the fewest number of multiplications) in which to perform the matrix multiplications to compute $A_1, A_2, ..., A_n$.

At each step, choose the smallest remaining dimension (from among $d_1, ..., dn - 1$) and multiply two adjacent matrices that share the same dimension.

State True/False:

The above strategy will always minimize the number of multiplications necessary to multiply the matrix chain $A_1, ..., A_n$.

ANS: False.

Consider the following three matrices: M_1, M_2, M_3 with dimensions $1 \times 1, 1 \times 10, 10 \times 2$. Following the above strategy, we would end up multiplying $(M_1 \times M_2) \times M_3$ for a total of 30 multiplications. However, if we multiply $M_1 \times (M_2 \times M_3)$, we would have 22 multiplications.