

Fundamentals of Algorithms (cs502)
Assignment No.5

Total Marks: 20
Due Date: 13th of July, 2010

Objectives

Objective of this assignment is to make students able to understand and implement Dynamic Programming, Greedy Algorithms and basic concepts relating to Graphs.
The topics covered in this assignment are,

- a. Coin Change Problem
- b. Huffman Encoding Problem
- c. Activity Selection Problem
- d. Basic Definitions of Graphs

Important Instructions

- a. More than one correct solution of a single question is possible
- b. Please do your own work, avoid copying solutions of questions from internet or any other resource.

Question No.1

Coin Change Problem:

Suppose a country ABC's currency has the following coins,

- a. 3 cent
- b. 6 cent
- c. 9 cent

- 1. You have to find the solution of coin change problem using Dynamic Programming approach for amount 12. [Your solution should have minimum number of coins].**
- 2. Find the solution of the above problem using Greedy Algorithm's approach.**

Hints:

- Your Table will have coins (denominations) as rows and values from 1 to 12 as columns. This problem is very similar to knapsack problem with the difference that here value is count of coins and we will select minimum coins count, you can see further details in handouts and video lecture.
- Since we want to minimize the number of coins used,
 $C[i, j] = \min(C[i - 1, j], 1 + C[i, j - d_i])$
- We will find the coins to be used in similar manner using separate table as we found in Knapsack problem.
- For Greedy Approach you will take coin of highest value first.

Question No.2

Huffman Encoding:

1. Suppose our text include only characters A, B, C, D, E and f and you have to encode the following string,

“AAABBDDEEEBABAABBB”

Encode this string using Huffman Encoding and showing all steps including finding probability of each character, making Optimal Huffman Encoding Tree and finding codes for each character. Also show the percentage compression achieved using Huffman Encoding for this string if original string was stored in the form of ASCII characters (8 bits for each character).

2. We can construct Huffman Encoding Tree using many ways; describe optimal way of constructing it.

Hints:

- Probability of each character will be equal to count of occurrence of that character divided by total characters occurrence count.
- Consult pseudo code for constructing Huffman Encoding Tree described in lecture to find optimal way of constructing Huffman Encoding Tree.

Question No.3

Activity Selection Problem:

While studying Activity Selection Problem we realized that sometimes we are able to find some heuristic (greedy algorithm) that help us to find one of the optimal solutions very efficiently. The heuristic was,

“We sorted the given activities on the base of their final times, using the fact that all activities in an acceptable schedule should have start time greater than or equal to finish time of previous activity. Then we scanned through all activities and keep on adding activities having start time greater than or equal to finish time of previous activity (note that first activity will always be in an acceptable schedule and our greedy algorithm will always result in same optimal solution (set of compatible activities))”

Consider the following activities table sorted on the base of their finish times

i	1	2	3	4	5	6	7	8
s _i	1	3	2	4	3	5	6	8
f _i	3	4	6	8	8	9	10	11

1. Find the optimal schedule using Dynamic Programming.

2. Find the optimal schedule using Greedy Approach.

Hints:

- We will add two pseudo activities a0 and a9 on both sides of these eighth activities because,

We need to compare the starting time of a1 with finish time of previous activity a0, similarly we need to compare the final time of last activity with starting time of its next activity. So we take finish time of a0 as $f_0 = 0$ and start time of a9 as $s_9 = \text{infinite}$, as shown in table below

i	0	1	2	3	4	5	6	7	8	9
s _i		1	3	2	4	3	5	6	8	infinite
f _i	0	3	4	6	8	8	9	10	11	

The Dynamic Programming Table structure for this problem will as follows,

		i →									
		0	1	2	3	4	5	6	7	8	9
j ↓	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Only shaded cells will be filled as for other cells $i \geq j$ (As activities are sorted on the base of finish time so higher number activities can not come before lower number activities hence un-shaded cells will be zero as they can never come in compatible schedule).

Cells will be filled as follows,

For any cell $[i, j]$, we will write the count of activities that are compatible with those activities (lie between time interval $[f_i, s_j]$, i.e **they start after activity i finishes and finish before activity j starts.**

The formula to calculate any cell value will be,

$$C[i, j] = \begin{cases} 0 & \text{if } S[i, j] = \phi \\ C[i, j] = \max [C[i, k] + C[k, j] + 1] & i < k < j \quad fS[i, j] \neq \phi \end{cases}$$

It means we will have to look the activities table to find that any activity between i and j is compatible with them if so we will write value according to above formula and we will write activity no. in k table.

- The diagonal cell entries will be zero as for these cells $i = j$
- For cells corresponding to two consecutive activities like $C[0,1]$, $C[1,2]$, $C[2,3]$, and so on... We will have $S_{ij} = \phi$, from this hint you can easily judge their values from above equation.
- After these cells as we will move towards top right corner we will have one or more activities between i and j and we will check whether any of them is compatible with activity i and j and we will find values for C and K table.

For cell C [0,2]

Our k can be 1 only [$i < k < j$]

So see whether activity 1 is compatible with 0 and 2, answer is Yes, as from table of activities

$a_0 \rightarrow a_1 \rightarrow a_2$

is acceptable as,

$f_0 \leq s_1 < f_1 \leq s_2 < f_2$

$0 \leq 1 < 3 \leq 3 < 4$

In simple words a_1 is compatible with a_0 and a_2 [can be run between a_1 and a_2]

$C[0,2] = C[0,1] + C[1,2] + 1 = 0 + 0 + 1 = 1$

$K = 1$

$C[0,2]$ value will be filled in C table and k value will be filled in K table.

For cell C [0,3]

Our k can be 1 or 2 [$i < k < j$]

So first see whether activity 1 is compatible with 0 and 3, answer is No as 3 start time is 2 and 1 finish time is 3 so they can not be compatible. Now check same for activity 2, answer is again No, so

$S_{ij} = \phi$, and above row of formula will be satisfied.

Question No.4

Graphs:

Give the answer of following questions:

- a. Difference between a Tree and a Graph ?
- b. Is every Graph a Tree ?
- c. Is every Tree a Graph ?
- d. What is difference between simple and complex cycles in a Graph ?
- e. What is Degree of a Graph?
- f. Is Binary Tree a graph if so then what is its degree?

Assignment Uploading Instructions:

Upload single doc file having solution of all these four questions.