SWE2016-44 Algorithms

Homework #4 (Due: December 1, 2019)

Professor: Joon Hee Choi *TA: Junsung Cho*

Problem 1 Dynamic Programming Given an array of n positive integers, find the sum of maximum sum subsequence such that the integers in the subsequence are sorted in increasing order. (This is a variation of Longest Increasing Subsequence problem.)

(a) Complete the the pseudo code shown below for the maxSumIS function. (15 pts)

maxSumIS(arr, n)

```
 1: initialize i, j, mpis[n], max=0
 2: /* Initialize mpis values for all indices */
 3: for(i = 0 to n-1) {
 4:   mpis[i] = arr[i]
 5: }
 6:
 7: /* Compute optimized MPIS values considering
 8:    every element as ending element of sequence */
 9: for(i = 1 to n-1) {
10:   for(j = 0 to n-1) {

11:     if(_________________________ and _________________________) {

12:       ___________________________________________________
13:     }
14:   }
15: }
16:
17: return max(mpis)
```

(b) Given an array Arr=[1, 101, 2, 3, 100, 4, 5], write down the values of mpis[] after running the code of Problem 1-(a) and the finally returned value. (10 pts)

Arr[]	1	101	2	3	100	4	5
mpis[]							

Final value:

Problem 2 Huffman Tree Suppose we are given the following table of letter frequencies:

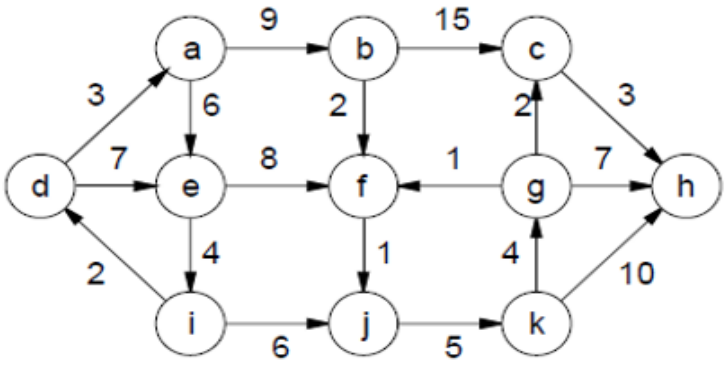
Value	A	B	C	D	E	F
Frequency	5	25	7	15	4	12

(a) Draw the Huffman coding tree. (15 pts)

(b) Write down the code of each character. (10 pts)

Value	A	B	C	D	E	F
Code						

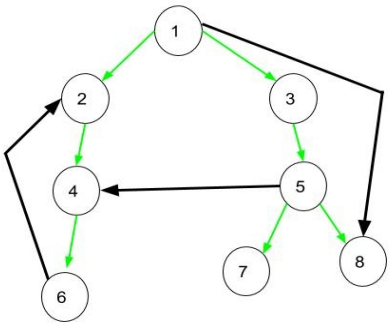
Problem 3 Graph Algorithm Consider the following weighted directed graph G:



In the following problems, you should assume that the elements of vertices[G] are ordered alphabetically, and that the vertices within each adjacency list are also ordered alphabetically.

(a) 1) Draw the tree that is induced by performing a depth-first-search **starting at node d** and **accessing alphabetically**. Label each vertex by its discovery and finish times. 2) Using dotted lines, draw the edges of G that are not in the depth-first tree and label them as forward edges (F), back edges (B), or cross edges (C). (15 pts)

Example



- forward edge: Edge from 1 to 8
- back edge: Edge from 6 to 2
- cross edge: Edge from 5 to 4

(b) Consider the directed acyclic graph (DAG) G' that is the subgraph of G that results from removing from G all edges labeled as back edges (B) in part (a). Give a topological sort of G' **starting at node d** and **accessing alphabetically**. (10 pts)

(c) Draw the tree that is induced by performing breadth-first search on G **starting at node d** and **accessing alphabetically**. (10 pts)

(d) Use Dijkstra's algorithm to determine the shortest paths from d to each of the other vertices. Draw the shortest path tree that is induced by running Dijkstra's algorithm, and annotate each vertex by its shortest path distance from d. (15 pts)