Homework #2

Homework exercises should be done individually. You write your solutions on paper by yourself, scan (or photo capture through a mobile application such as CamScanner) and submit them as a single .pdf file. Your solutions have to be handwritten.

Solutions must be submitted electronically as .pdf file before **2.00 pm on May 30**. **No credit will be given to solutions obtained verbatim from the Internet or other sources.**

1.(20p) Given a sequence of n integers $x_1, x_2, \dots, x_n$, the product-sum is defined as the largest sum of multiplications of adjacent elements in the sequence. Each element can be matched (multiplied) at most one of its neighbors.

For example:

for the sequence 1, 2, 3, 1; the product-sum will be $1 + 2\times3 + 1 = 8$
for the sequence 1,4,3,2,3; the product-sum will be $1 + 4\times3 + 2\times3 = 19$

Your task is to design a dynamic programming algorithm that computes the product-sum of a given sequence of integers.

a) (10p) define a subproblem, construct the recurrence relation (opt(i) = ?), and identify base case(s)

b) (10p) use your student number as the input, and build a table for the solutions of the subproblems on your input

Assume your id is 14290519. The table for the input 1, 4, 2, 9, 0, 5, 1, 9 will be as follows :

opt(1)	**opt(2)**	**opt(3)**	**opt(4)**	**opt(5)**	**opt(6)**	**opt(7)**	**opt(8)**
			23			29	

Note that for this example, I just filled two entries. But you must fill all the entries in the corresponding table.

2.(20p) For the question, first construct a string using your student id as follows ('14290519' will be used for the following string):

- take the square of your id

  $14290519^2 = 204218933289361$

- multiply the square with the number 123456789 (fixed for all the students)

  25212213756909716921829

- write the corresponding letters from the table in place of the numbers you obtained above in order to get your string (Note that all the students must use the following table to transform the numbers to the letters)

0	1	2	3	4	5	6	7	8	9
A	B	C	D	E	F	G	H	K	L

your string = "CFCBCCBDHFGLALHBGLCBKCL"

a) Find the frequencies of the characters in the string, and construct the Huffman tree corresponding to these frequencies

b) Based on the Huffman tree you have obtained, determine the Huffman codes of the characters, and calculate the cost of the encoding for your string

3.(25p) For the questions, first construct a weighted directed graph using your student id as follows ('14290519' will be used for the following graph):

- take the square of your id

  $14290519^2 = 204218933289361$

- remove all the zeros from the resulting number

  $204218933289361 \longrightarrow 24218933289361$

- cut out the first 9 numbers

  242189332

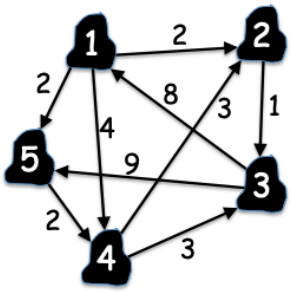
- fill the empty entries in the following adjacency weighted matrix with the obtained ten numbers, with the following order 'from the first row to the last row, and at each row, from left to right.

	1	2	3	4	5
1	0		∞		
2	∞	0		∞	∞
3		∞	0	∞	
4	∞			0	∞
5	∞	∞	∞		0

→

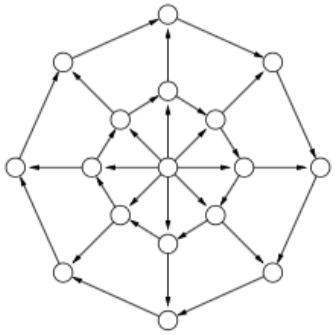
	1	2	3	4	5
1	0	2	∞	4	2
2	∞	0	1	∞	∞
3	8	∞	0	∞	9
4	∞	3	3	0	∞
5	∞	∞	∞	2	0

- construct the corresponding graph



Assume you are using the Floyd-Warshall algorithm to find the shortest paths between every pair of vertices, and $D^{(5)}$ is the final matrix output by the algorithm. Just write the matrices $D^{(3)}$ and $D^{(4)}$.

4.(15p) A wheel graph is a directed graph of the following form, i.e. a wheel graph consists of a center vertex c with k outgoing 'spokes' of s outward oriented edges at each circle; furthermore, all the spokes at each circle are connected to form a directed cycle, and all cycles are oriented the same way (k = 8 and s = 2 for the following figure):



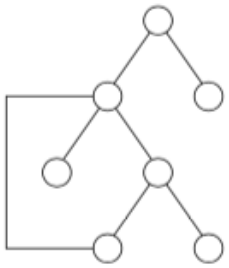
Assume you are given a weighted loop graph (each edge has a positive integer weight).

a) How long does it take for Dijkstra's algorithm to output the shortest paths from center to all other nodes in a given wheel graph (as a function of n where n is the number of vertices in the graph?

b) Design an efficient algorithm (better than Dijkstra) for finding the shortest paths from center to all other nodes in a given wheel graph. To get full credit, you need to argue the running time.

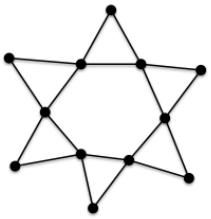
5.(20p) For the following two questions, you just pick one of them to answer.

Option 1. Consider an undirected graph which is formed by taking a binary tree and adding an edge from exactly one of the leaves to another node in the tree. We call such a graph a loop graph. An example of a loop graph could be the following one:



Assume you are given a weighted loop graph (each edge has a positive integer weight). Design an efficient algorithm (better than Kruskal and Prim) for finding the minimum spanning tree of a given star graph. To get full credit, you need to argue the running time.

Option 2. A star graph is an undirected graph of the following form, i.e. let n be the number of vertices, a star graph consists of a center cycle with n/2 vertices where each pair of connected vertices are also connected to a third vertex:



Assume you are given a weighted star graph (each edge has a positive integer weight). Design an efficient algorithm (better than Kruskal and Prim) for finding the minimum spanning tree of a given star graph. To get full credit, you need to argue the running time.