

CS502 Fundamentals of Algorithms

Final Term Examination – Spring 2005

Time Allowed: 150 Minutes

Please read the following instructions carefully before attempting any of the questions:

1. The duration of this examination is 150 Mins.
2. This examination is closed book, closed notes, closed neighbors.
3. Do not ask any questions about the contents of this examination from anyone.
 - a. If you think that there is something wrong with any of the questions, attempt it to the best of your understanding.
 - b. If you believe that some essential piece of information is missing, make an appropriate assumption and use it to solve the problem.
4. Some of the examination consists of multiple-choice questions. Choose only one choice as your answer.
 - a. If you believe that two (or more) of the choices are correct for a particular question, choose the best one.
 - b. On the other hand, if you believe that all of the choices provided for a particular question are wrong then select the one that appears to you as being the least wrong.

****WARNING: Please note that Virtual University takes serious action against unfair means. Anyone found involved in cheating will get an `F` grade in this course.**

Some results you may need:

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}, \quad \sum_{i=1}^n i^2 = \frac{2n^3 + 3n^2 + n}{6}, \quad \sum_{i=1}^n x^i = \frac{x^{(n+1)} - 1}{x - 1}$$

Total Marks: 95

Total Questions: 08

Question No. 1

Marks : 05

Best and worst case times of an algorithm may be same.

- ☐ True
- ☐ False

Question No. 2

Marks : 05

Can an adjacency matrix for a directed graph ever not be square in shape?

- ☐ Yes
- ☐ No

Question No. 3

Marks : 05

If an algorithm has a complexity of $2n^2 + 4n + 3$ for some model of computation (some set of assumptions) and some complexity measures (such as number of comparison operations) we could say that it has complexity

- ☐ (a) $O(\log_2 n)$
- ☐ (b) $O(n^2)$
- ☐ (c) $O(2 + 4 + 3)$
- ☐ (d) all of the above
- ☐ (e) none of the above

Question No. 4

Marks : 05

You are given the task of laying down new railway lines which will connect all n cities. Thus for any pair of cities, you will end up with track connecting them. Note that two routes may share the same track; track laid between Lahore and Islamabad can be used to travel in both directions. your goal is to use the minimum amount of track. How would you achieve the goal now? (Note : consider the scenario carefully and name only the best suited algorithm)

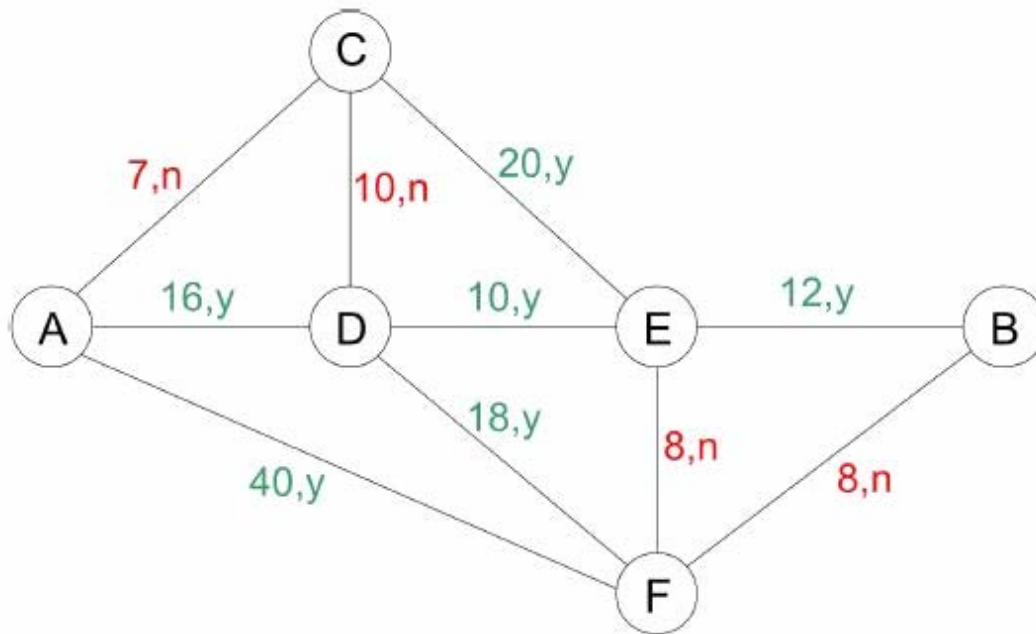
- ☐ 1 Dijkstra's algorithm
- ☐ 2 Prims algorithm
- ☐ 3 Floyd Warshall algorithm
- ☐ 4 Bellman Ford algorithm.

Question No. 5

Marks : 25

You land at Heathrow airport for a vacation in London. Your hotel located in Essex, a suburb of London. You will need to take the train to get to your hotel. You have a map that shows the train lines and the fares between various points in London. You would like to minimize the fare you need to pay to travel from the airport to your hotel. To make matters complicated, the airline you traveled on gave you a discount coupon that can be used on *one* train journey. The coupon can be used only once. The coupon will get you 50% discount on the train fare. The coupon also lists which train lines the coupon can be used on to get the 50% discount. Solve the problem on the following sample train lines map (node A is the airport, node B is Essex where the hotel is). The notation 40,y means the fare is 40 and discount is available. If the coupon is used on this line, the fare would be 20. Describe the algorithm you have used to determine the train route that minimizes the total fare.

[25 pts]



Question No. 6

Marks : 10

Solve the following recurrence using iteration method (show intermediate steps) [10 pts]

$$T(n) = n + 2T(n/2)$$

$$T(1) = 1$$

Question No. 7

Marks : 25

Recall that a dynamic programming solution to the 0-1 knapsack problem can be derived from the following recurrence formula for $c[i, w]$, the value of the solution for items $1, \dots, i$ and maximum weight w .

$$c[i, w] = \begin{cases} 0 & \text{if } i = 0 \text{ or } w = 0, \\ c[i-1, w] & \text{if } w_i > w, \\ \max(v_i + c[i-1, w - w_i], c[i-1, w]) & \text{if } i > 0 \text{ and } w \geq w_i. \end{cases}$$

>:

The algorithm based on this recurrence takes as inputs the maximum weight W , the number of items n , and the two sequences $v = \langle v_1, v_2, \dots, v_n \rangle$ and $w = \langle w_1, w_2, \dots, w_n \rangle$. It stores the $c[i, j]$ values in a table $c[0..n, 0..W]$. At the end of the computation, $c[n, W]$ contains the maximum value the thief can take.

In the following example the inputs are $n = 6, W = 8$, with values v_i and weights w_i :

i	1	2	3	4	5	6
v_i	8	6	5	9	9	5
w_i	3	4	2	4	3	4

From these inputs the following c table is computed by the algorithm:

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	8	8	8	8	8	8
2	0	0	0	8	8	8	8	14	14
3	0	0	5	8	8	13	13	14	14
4	0	0	5	8	9	13	14	17	17
5	0	0	5	9	9	14	17	18	22
6	0	0	5	9	9	14	17	18	22

The last part of the algorithm uses this table to determine which items the thief should take to achieve the maximum value, 22.

(a) Describe this last part of the algorithm: how, in general, it determines the items to be taken. (Note : in maximum three lines.)

(b) For the above example, list the items to take (i.e., list their indices).

[25 pts]

Question No. 8**Marks : 15**

Run DFS sweep and topological sort on the directed graph defined by the following adjacency matrix.

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

[15 pts]
