CS345A: Algorithms -II
Users Online : 7

End Semester Exam

Submitted on 22/11/2021 11:09

Instructions

- Exam opens at: 22/11/2021 09:00
- You are given an extra 10 minutes after due time to submit your exam.
- However, please note that any submissions made after the due time are marked as late submissions.

End Semester Exam

Question:

Instructions:

This exam has **2 problems**. Each problem consists of 2 or more questions. Please attempt exactly one question per problem. If you attempt more than 1 questions of a problem, you will get 0 marks for that problem.

Problem 1: Attempt exactly one of the following 2 questions.

Question 1: *Destroying bridges* (*marks*=15)

Let G=(V,E) be an undirected connected graph on n vertices and m edges given in the form of the adjacency lists. Vertices are numbered fron to n. An edge in G is called a bridge if its removal will disconnect G, that is, its removal will result in 2 connected components. You are told tha there are t bridges in G and an adversary will delete these bridges one by one. However, you have no prior knowledge about the order in whi these bridges will be deleted. Your aim is to maintain an array A[1..n] that has to satisfy the following property after the deletion of the first i bridges by the adversary for each 1≤ i ≤ t, and for each 1≤j,k ≤n :

A[j]=A[k] if and only if vertex j and vertex k belong to the same connected component in the graph obtained after the removal of the first i bridges from G by the adversary.

You have to design an algorithm that must take total O(m log t) time for maintaining A for any sequence of the deletions of all t bridges. The additional space used by your algorithm must not exceed O(m).

Note: You need to give the details of the algorithm (9 *marks*). In addition, you need to provide suitable arguments that establish the O(m log t bound on the total time (6 *marks*). There is no need to prove correctness of the algorithm.

Question 2: (Maximum share profit) (*marks*=8)

There is an array A[1..n] storing the prices of a share on n consecutive days. If we buy a share on ith day and sell it on jth day, we incur a profit A[j]-A[i] for any 1≤i ≤ j ≤n. Note that it may be loss as well if A[j] < A[i]. Our aim is to compute the maximum possible profit we can obtain by buying a share on some day and selling it on some (same/later) day. The following is the pseudo-code of a divide and conquer algorithm to compute this maximum profit for a period [i,j] where 1≤i ≤ j ≤n. Fill in the blanks suitably. Note that there are total 8 blanks. You may use *min* c *max* operators according to your convenience with any constant number of operands. For example *max*(*a,b*) represents the bigger of the two numbers *a* and *b; min(a,b,c)* represents the smallest number among *a,b* and *c*. However, you must not add any extra statement in the pseudo-code.

HighestProfit(A, i, j)

{

If (i=j) return (A[i], A[j],0)

else

{ k ← ⌊(i+j)/2⌋;

(low_L, high_L, profit_L) ← HighestProfit(A, ... , ...);

CS345A: Algorithms -II
Users Online : 7

profit ← ....;

return (low, high, profit);

}

The time complexity of this algorithm is ....

Problem 2: Attempt exactly one of the following 3 questions.

Question 1 (Approximation algorithm for maximum number of paths) (marks=15)

Let G=(V,E) be a directed graph on n=|V| vertices and m=|E| edges. x and y are any two designated vertices in G. We wish to compute the largest set S of paths from x to y such that no vertex from the set V\{x,y} appears in 2 or more paths in S. Consider the following algorithm designed to solve this problem approximately.

Algorithm-Approx(G, x,y)

S ← ∅;

While(there is any path from x to y in G) do

{

Let P be the shortest path from x to y (break any tie, if exists, arbitrarily);

Add P to S;

Remove all vertices of P, except {x,y}, from G;

}

Return S.

Which of the following statements is true about *Algorithm-Approx*?:

1. The number of paths reported by *Algorithm-Approx* on every graph G is at least 1/2 of the number of paths in the optimal solution of G.

2. The number of paths reported by *Algorithm-Approx* on a graph G can be less than 1/2 of the number of paths in the optimal solution of G. However, the number of paths reported by *Algorithm-Approx* on every graph G is at least 1/($\log_e$ n) of the number of paths in the optimal solution of G.

3. The number of paths reported by *Algorithm-Approx* on a graph G can be less than (3/n) of the number of paths in the optimal solution of G

Note: You must justify your answer formally. No marks for guess work. Very few marks will be awarded in case the justification is incomplete vague or informal.

Question 2 (*Straight from the practice sheet***)** (marks=9)

Let G=(V,E) be a directed graph on n=|V| vertices and m=|E| edges with a designated source vertex s and designated sink vertex t. The capaci of each edge is 1. Furthermore, the distance from s to t is $n^{2/3}$. Prove formally that the Ford-Fulkerson algorithm will take $O(mn^{2/3})$ to compute the maximum flow from s to t.

Note: You must give complete details of the formal analysis. Otherwise, there will be heavy penalty.

Question 3: Binomial Heap (marks=6)

CS345A: Algorithms -II
Users Online : 7

Uploaded Files:

CS345_180771.pdf

Grades:

Marks: Not graded