Q1 14

B5DEBD48-EBC5-4804-BBEB-5A697CB11B2E

csci-570-fa-22-exam-1

#159 2 of 12

1) 16 pts
Mark the following statements as **TRUE** or **FALSE** by circling the correct answer.
No need to provide any justification.

[TRUE/FALSE]
There exists an instance of the Stable Matching problem in which two men have the same best valid partner. False

[TRUE/FALSE]
Prim's algorithm is not guaranteed to return a correct solution for graphs with negative weights.

[TRUE/FALSE]
If $T(n) = 4T(n/2) + 8n^2$, then $T(n) = O(n^3)$

[TRUE/FALSE]
For a weighted connected undirected graph G with positive weights, if the edge e is not part of any MST of G, then e must be the unique maximum weight edge of some cycle in G. T

[TRUE/FALSE]
If a binomial heap consists of the 3 binomial trees B_0, B_1, and B_3, then after 4 Extract_Min operations, the binomial heap will consist of the following 3 trees: B_0, B_1, and B_2 T

[TRUE/FALSE] X
For any cycle in a weighted connected undirected graph G with positive weights, if the cycle has a unique least-weight edge, then that edge is in some minimum spanning tree of G.

[TRUE/FALSE]
If algorithm A has a worst-case running time of $\Theta(n^3)$ and algorithm B has a worst-case running time of $\Theta(n^2)$, then algorithm B always runs faster than algorithm A on the same input.

[TRUE/FALSE]
In every undirected graph, there exists at least one path between every pair of vertices.

Q2 6

A2D901FA-BCE2-49B7-A30F-308FFB4863DC

csci-570-fa-22-exam-1

#159 3 of 12

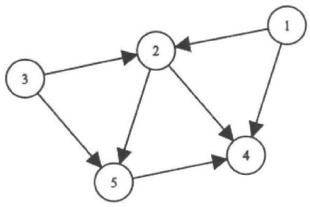
2) 12 pts
Circle ALL correct answers (no partial credit when missing some of the correct answers or circling some of the incorrect answers). No need to provide any justification.

i- Which of the following contradicts the statement, "The worst-case running time of the algorithm is $\Omega(n^2)$"? (3 pts)
(a) The algorithm runs in $O(1)$ steps on some types of input.
(b) For no input does the algorithm run in $O(n)$ steps.
(c) The worst-case running time is $O(n \log n)$.
(d) The worst-case running time is $O(2^n)$.
(e) The worst-case running time is $\Omega(n^3)$.

ii- Consider a binary max heap represented as an array [10, 9, 6, 8, 7, 4, 1, 2, 3]. We perform an Extract_Max followed by Decrease_Key(9,5) [meaning that the element with key value 9 will now have a key value of 5] on this heap. Which of these represents the new state of the heap? (3 pts)
(a) [8, 6, 7, 5, 4, 3, 1, 2]
(b) [8, 7, 6, 3, 5, 4, 1, 2]
(c) [8, 6, 5, 7, 4, 1, 2, 3]
(d) [8, 7, 6, 3, 5, 4, 2, 1]

iii- Which of the following algorithms solve the Minimum Spanning Tree problem?
(3 pts)
(a) Kruskal's Algorithm
(b) Dijkstra's Algorithm
(c) Strassen's Algorithm
(d) Prim's Algorithm

iv- Which of the following is/are valid topological sorting(s) for the given DAG?
(3 pts)



1) 1,3,2,5,4
2) 1,3,5,2,4
3) 3,1,2,5,4
4) 3,1,5,2,4
5) 4,5,2,3,1

9AD202EF-0A26-45EC-9925-3A470B33F42F
csci-570-fa-22-exam-1
#159 5 of 12

Q3 6

3) 8 pts
For the given recurrence equations, solve for *T(n)* if it can be found using the Master Method (make sure to show which case applies and why). Else, indicate that the Master Method does not apply and explain why.

i) $T(n) = 8T(n/2) + n\log n - 1000n$

$a=8 \quad b=2 \quad \therefore n^{\log_b a} = n^{\log_2 8} = n^3$

$f(n) = (n\log n - 1000n) = O(n^{3-\varepsilon})$

$\therefore T(n) = \Theta(n^3)$ Case 1 of Master Theorem.

ii) $T(n) = 2T(n/2) + n^3(\log n)^3$

$a=2 \quad b=2 \quad \therefore n^{\log_b a} = n^{\log_2 2} = n^1$

$f(n) = (n^3(\log n)^3) = \cancel{\Omega(n^{1+\varepsilon})} = \cancel{\Theta(n^3 \lg^3 n)} \cancel{\Rightarrow T(n) = \Theta(n^3 \log n)}$

~~Using Case 3~~ ~~generalized one~~ :- $\cancel{T(n) = \Theta(n^3(\log n)^4)}$ Master theorem won't apply here, as f(n) = $n^3\log^3 n$ which doesn't fall in any generalized form.

iii) $T(n) = 4T(n/2) + n^2(\log n)^2$

$a = 4 \quad b=2 \quad \therefore n^{\log_b a} = n^{\log_2 4} = n^2$

$f(n) = n^2(\log n)^2 = \cancel{\Theta(n^{2})} \ \Theta(n^2 \lg^2 n)$

$\therefore T(n) = \Theta(n^2(\log n)^3)$ using ~~case~~ generalized case ~~vers~~ of 3 where f(n) has a log n term

iv) $T(n) = 4T(n/2) - n^4(\log n)^4$

$a = 4 \quad b = 2$

but f(n) is negative and so master's theorem cannot be applied.

Q4 8

C0F85772-DB1B-44F7-B826-FB5869B8D2BB

csci-570-fa-22-exam-1

#159 6 of 12

4) 12 pts

A student wants to insert n elements into an empty binary heap. The student also wants to backup this heap after every fixed number of insertions. Unfortunately, the backup operation is quite costly: each backup operation takes $\Theta(n)$ time (no matter how many elements are currently in the heap).

a) What is the amortized cost of the insertion operation if backups are performed after every n/10 insertions? You must show all your work. (6 pts)

We charge insertion operation 11 and backup 0 with actual costs of 1 and n respectively. After every insertion, we will first pay the operation 1 and leaving 10 as credit. So, after $\frac{n}{10}$ insertions, $\frac{n}{10} \times 10 = n$ credits in the bank. This n credits will be used by the backup and so the amortized cost = O(1).

-1 wrong binary heap runtime -1

-2 Wrong answer -2

b) What is the amortized cost of the insertion operation if backups are performed after every 10 insertions? You must show all your work. (6 pts)

We charge insertion $(\frac{n}{10}+1)$ and backup 0 with actual cost of 1 and n respectively. We will be paying the insertion operation 1 and therefore after 10 insertions we will have $(\frac{n}{10}+1) \times 10 = n+10$ credits. Over here the amortized cost will be O(n).

-1 wrong binary heap runtime -1

Q5 10

5) 20 pts
The transportation network of bus, train, and airplane routes in California can be represented as a weighted connected undirected graph $G(V,E)$ with positive weights, where each vertex $v \in V$ represents a city in California, each edge $e \in E$ represents a transportation route between two cities, and each edge weight $w(e)$ represents the length of time needed to travel via the transportation route e. Each edge $e \in E$ is either a bus route, train route, or airplane route. Denote the set of bus routes as $E_B \subseteq E$, the set of train routes as $E_T \subseteq E$, and the set of airplane routes as $E_A \subseteq E$. Note that there may be more than one transportation route type connecting two cities. For example, there may be a train route and a bus route between the same two cities.

Use Dijkstra +5

a) Design an efficient algorithm to compute [illegible] route from a given city s to a given city t that never trave[illegible] on twice consecutively. For example, if we take a train [illegible] train out of city j. (17 pts)

Do some modification to Dijsktra +2

We will run Dijkstra's w[illegible] will be selecting the [illegible] city one set of routes and co[illegible] the undirected edges to directed. We will then run Dijkstra's from city s to city t and compute the length of time. We won't be using the same set of routes while computing from t to s and so, the set of routes which we select initially will all be directed in the same direction and the rest of the sets of routes in the opposite. In this way we can make sure that the same mode of transportation won't be used twice. But if this is only one ~~route~~ transportation route connecting 2 cities, then there is no way to compute the length and so the solution doesn't exist.

b) Analyze the worst-case time complexity of your algorithm. (3 pts)

Correct (b) +3

Selecting any set of routes can be done in constant time. We are running Dijkstra's ~~once~~ so the complexity would be O(e log v). Changing the edges from undirected to directed will also take time less than O(e log v). So worst case time complexity = O(e log v)

Q6 19

A8C8B72F-360B-49A2-A939-494600715ADF

csci-570-fa-22-exam-1

#159 8 of 12

6) 22 pts
Assume there are n TAs for a graduate-level CS course. The TA availability on Mondays is provided in the form of two arrays S[1..n] and E[1..n]. For example, for the first TA in the list, S[1] = 8 and E[1]=13 indicates that this TA will be available from 8 AM to 1 PM.

a) Design an algorithm that returns the minimum number of TAs required so that there is at least one TA available from 8 AM to 8 PM on Mondays. (12 pts)

We will be sorting array E or select the index with the least value thus ensuring earliest endtime. We denote E array as endtime array and array S as start time array. Once such an element ~~index~~ is selected we will then remove all the other elements that overlap with it. We will then select the element which is next in the array. By doing so, we are ensuring that there is atleast one TA available from 8 AM to 8 PM on Mondays and by removing the TAs who's end timings overlap we ensure selecting the minimum number.

Part b and c of this problem are on the next page

The student constructed a set of TAs by iteratively adding the TAs to their constructed set, but did so incorrectly. -3

b) Analyze the worst-case time complexity of your algorithm. (3 pts)

Sorting the array E in the order $E(i) \leq E(j)$ where $i<j$, this takes $O(n \lg n)$. Selecting requests in order of increasing $E[i]$ by always selecting the first element. Then iterating through the intervals until we encounter $S[j] \geq E[i]$. This takes $O(n)$. Thus the overall complexity would be $O(n \lg n)$

c) Prove that your algorithm is correct. (7 pts)

We will prove the correctness by showing that we selected the minimum number of TA's by eliminating the one's which overlap. this was already done by us as we removed all the instances of elements which overlapped with $E[i]$. Next we show that our algorithm is the returning the same number as the optimal(O). We denote B as our algorithm and O as the optimal one. The requests in B as $i, \dots, k$ and that in O as $j, \dots j_m$. we will prove that for all indices $r \leq k$ we have $E(i_r) \leq E(j_r)$ in a way showing that our solution is always going to stay ahead than the optimal one. By mathematical induction, Basecase $B(i_1)$ stays ahead than $O(j_1)$. Inductive hypothesis:- We say that there exists $[j_{r+1}]$ such that $E[j_{r+1}]$ is compatible with $[i_r]$ and so they are a part of the optimal solution. But we know that (i_r) finishes no later than (j_r) and therefore i_r should also be compatible with j_{r+1}. But we will be picking i_{r+1} and so $E(i_{r+1}) \leq E(j_{r+1})$ and we say that $|B| = |O|$. Proving that the number of TA's in B and O are equal.

Q7 4

608DB1C6-4F7A-48CD-99E4-29B254A64717
csci-570-fa-22-exam-1
#159 10 of 12

7) 10 pts
Prove or disprove the following statement:

For a weighted connected undirected graph G(V,E) with positive weights, if for all edges e ∈ E there exists at most one other edge e' ∈ E with the same weight, then G has at most two distinct minimum spanning trees.

If for all edges e ∈ E there are edges e' ∈ E with same weight, then there will only be one minimum spanning tree (i.e all the edges would be having the same weight). But if for all the edges e ∈ E there are no edges e' ∈ E with the same ~~ea~~ weight then, there also will ~~ca~~ be only minimum spanning tree where the weights of edges would be distinct. Thus this statement is false.

wrong reasoning No counter example provided -6

Additional Space

724B8B2F-55D3-4DD7-8044-887F429479FD

csci-570-fa-22-exam-1

#159 12 of 12

Additional Space