# CS570 Fall 2022: Analysis of Algorithms Exam I

| | Points | | Points |
|---|---|---|---|
| Problem 1 | 16 | Problem 5 | 20 |
| Problem 2 | 12 | Problem 6 | 22 |
| Problem 3 | 8 | Problem 7 | 10 |
| Problem 4 | 12 | | |
| | **Total** | **100** | |

Instructions:

1. This is a 2-hr exam. Open book and notes. No electronic devices or internet access.
2. If a description to an algorithm or a proof is required, please limit your description or proof to within 150 words, preferably not exceeding the space allotted for that question.
3. No space other than the pages in the exam booklet will be scanned for grading.
4. If you require an additional page for a question, you can use the extra page provided within this booklet. However please indicate clearly that you are continuing the solution on the additional page.
5. Do not detach any sheets from the booklet. Detached sheets will not be scanned.
6. If using a pencil to write the answers, make sure you apply enough pressure, so your answers are readable in the scanned copy of your exam.
7. Do not write your answers in cursive scripts.
8. This exam is printed double sided. Check and use the back of each page.

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.

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

**[ TRUE/FALSE ]**
If $T(n) = 4T(n/2) + 8\, n^2$, 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 $G$.

**[ 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 $B_2$

**[ TRUE/FALSE ]**
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.

2) 12 pts

Circle ALL correct answers (no partial credit when missing some of the correct 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 $\Omega(n^3)$.

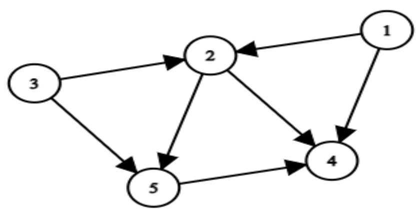
ii- Consider a binary max heap represented as an array [10, 9, 6, 8, 7, 4, 1, 2, 3]. We perform an Extract_Max followed by Decrease_Key(9,5) [meaning that the element with key value 9 will now have a key value of 5] on this heap. Which of these represents the new state of the heap? (3 pts)

(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

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

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

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

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

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? (6 pts)

b) What is the amortized cost of the insertion operation if backups are performed after every 10 insertions? (6 pts)

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.

a) Design an efficient algorithm to compute the length of time of the quickest route from a given city $s$ to a given city $t$ that never travels via the same mode of transportation twice consecutively. For example, if we take a train from city $i$ to city $j$, then we can't take a train out of city $j$. (17 pts)

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

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 (announced clarification: efficient) 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)

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

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

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* $e \in E$ *there exists at most one other edge* $e' \in E$ *with the same weight, then* G *has at most two distinct minimum spanning trees.*

Additional Space

Additional Space

Additional Space