CS570 Spring 2022: Analysis of Algorithms Exam I

	Points		Points
Problem 1	20	Problem 5	18
Problem 2	9	Problem 6	16
Problem 3	6	Problem 7	13
Problem 4	8	Problem 8	10
	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. 20 pts
   Mark the following statements as **TRUE** or **FALSE** by circling the correct answer. No need to provide any justification.

   i) [TRUE/FALSE]

   If we add *1* unit to the cost of the two lowest cost edges in the graph G, then the cost of the MST of G will increase by *2* units

e.g. the first 3 lowest cost edges could have the same cost

   ii) [TRUE/FALSE]

   Every weighted undirected graph has at least one MST

(If the graph is not connected, there is no MST)

   iii) [TRUE/FALSE]

   We say that an algorithm runs in $O(1)$ if it takes it constant time to run when the problem size $n=1$

   iv) [TRUE/FALSE]

   A binary tree with k levels has 2^k-1 nodes.

   v) [TRUE/FALSE]

   The worst-case time complexity of merge sort is $O(n^2)$

   vi) [TRUE/FALSE]

   The worst case runtime of binary search satisfies the recurrence relation $T(n) = 2T(n/2)+c$ where c is a constant

   vii) [TRUE/FALSE]

   If a directed acyclic graph with 4 nodes has a unique topological ordering (*ABCD*), then it must have at least 3 edges.

   viii) [TRUE/FALSE]

   Prim's algorithm cannot handle negative cost edges

ix) [TRUE/FALSE]

A binary max heap can be converted into a min heap by reversing the order of the elements in the heap array.

x) [TRUE/FALSE]

For $n>3$, a directed graph with n nodes and n edges can be strongly connected.

2. 9 pts

Circle ALL correct answers (no partial credit when missing some of the correct answers). No need to provide any justification.

i- If a binomial heap contains these three trees in the root list: B0, B1, and B3, after 2 Extract_min operations it will have the following trees in the root list. (3 pts)

a) a B0 and a B3

b) a B1 and a B2

c) a B0, a B1 and a B2

d) None of the above

ii- What's the solution to the recurrence $U(n) = 2U(n/2) + n \log n + 2n$. (3 pts)

a) $U(n) = \Theta(n \log n)$

b) $U(n) = \Theta(n \log^2 n)$

c) $U(n) = \Theta(n \log \log n)$

d) None of the above

iii- Let $G(V,E)$ be a weighted undirected connected graph. Which of the following are true? Choose all that are true. (3 pts)

(a) Minimum edge in a graph is always part of a MST.

(b) Minimum edge in a cycle is always part of a MST.

(c) Maximum edge in a cycle is never part of a MST.

(d) Maximum edge in a graph is never part of a MST.

3. 6 pts

For the given recurrence equations, solve for $T(n)$ if it can be found using the Master Method. Else, indicate that the Master Method does not apply.

Answer: each has 1.5 points

no explanation correct answer:0.75

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

a=1, b=2, f(n)=2^n (0.25 point)

$n^{log_2^1}$ =n^0=1 (0.25 point)

Case 3 Master Theorem: f(n)=$\Omega(nlog_2^{1+\epsilon})$,$2^{n-1} \le c2^n \Rightarrow$ T(n)=$\theta(2^n)$(1 point)

ii) $T(n) = 5T(n/5) + n \log n - 1000n$

a=5, b=5, f(n)= n log n – 1000n (0.25 point)

$n^{log_5^5}$ =n^1=n (0.25 point)

General case 2: f(n)=$\theta(n^{log_5^5}\ log^1 n)$=$\theta(n\ log\ n)$ $\Rightarrow$ T(n)=$\theta(nlog^2 n)$(1 point)

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

a=2, b=2, f(n)= $log^2 n$ (0.25 point)

$n^{log_2^2}$ =n^1=n (0.25 point)

Case 1: $log^2 n$ =$O(nlog_2^{2-\epsilon})$=> T(n)=$\theta(n)$(1 point)

iv) $T(n) = 49T(n/7) - n^2 \log n^2$

a=49, b=7, f(n)=$- n^2 \log n^2$ (1.5 point)

not applicable: (0.5 points)

problem is not a correct form to apply master theorem:T(n) = aT(n/b) + f(n)
The prerequisite regarding the positive f(n) is not satisfied. (1 point)

4. 8 pts

Analyze the worst-case complexity of the following code snippets and provide a tight upper bound for each case. No explanations necessary. Each part has 2 pts.

i)

```
k = 0
for i = 1 to n
    k = k + 1
endfor
```

ii)

```
k = 1
for i = 1 to n
    j = 1
    while j <= i
        k = k + 1
        j = j * 2
    endwhile
endfor
```

iii)

```
k = 570
i = 1
while i < n
    i = i * k
endwhile
```

iv)

```
x = 0
k = 57075
for i = 1 to min of (n, k)
    for j = 1 to log n
        x = x + k
    endfor
endfor
```

Solution:

i) O(n) 2 points

The problem involves n addition, therefore, the complexity is O(n)

Any other answer gets 0 points

ii) O(n log n) 2 points

If you answered O(log n!) you should still get 2 points (If you didn't get any point or you got only 1 point request a regrade, please)

iii) O($\log_k$ n) or O(log n) 2 points

The number of times that 1 can be multiplied by k before it gets greater than n is $\log_k n$

Any other answer gets 0 points

iv) O(log n) 2 points

The inner loop repeats for log(n) time

The outer loop is a constant number of iterations because k is constant

if you answered O(min(n,k)log n) you only get 1 point

if you answered as follows you get 1 point:

if n < k   O(n log n)

else: O(k log n)

5. 18 pts

During Spring break, which is m days long, USC is making its alumni park available to charitable organizations for fundraising events. However, only 1 event can be hosted on each day from day 1 to m. Each event lasts for exactly 1 day.

There are N prospective events. For each event i, there is a deadline D_i, denoting the last day on which it can be hosted, and expected funds F_i, denoting the funds it is expected to raise if it is hosted on or before its deadline. You cannot schedule an event after its deadline, i.e. event i can only be scheduled on days 1 through D_i only.

The objective is to create a schedule that will maximize the expected funds raised. Your schedule will assign n events to m days.

Note : n<=N and n<=m. (It may not be possible to schedule all events before their deadlines, in which case you will have to skip some events)

a. Consider the greedy algorithm that assigns n events to m days without any gaps and in increasing order of event deadline D_i (if by doing so an event i happens to land after its deadline *Di* then that event is skipped, i.e. not scheduled). Give a counterexample that shows this algorithm will not always yield an optimal solution. (5 pts)

b. Consider the greedy algorithm that assigns n events to m days without any gaps and in decreasing order of Expected Funds F_i (if by doing so an event i happens to land after its deadline *Di* then that event is skipped, i.e. not scheduled). Give a counterexample that shows this algorithm will not always yield an optimal solution. (5 pts)

c. Give an efficient greedy algorithm that will result in an optimal solution to this problem. No need to provide a proof of correctness. (8 pts)

A.

m=2, N=3, D = [1,2,3], F = [10,20,30]

Algorithm will give below day to event mapping

day 1 : event 1 : funds = 10

day 2 : event 2 : funds = 20

total funds raised 30

Optimal solution (Event 2, Event 3) 30+20 = 50.

B.

m=2, N=3, D = [1,2,3], F = [20,10,30]

Algorithm will give below day to event mapping

day 1 : event 3 : funds = 30

day 2 : can't schedule event 1 as its deadline is gone, hence schedule event 2 : funds = 10

total funds raised = 40

1 optimal solution (Event 1, Event 3) 20+30 = 50.

C.

Sort all events on decreasing order of expected funds.

Iteratively assign the event i to the last available day before or on its deadline. This day is now unavailable for other events. If no slot is available. Skip this event.

Other solutions may exist

Rubrics:

part(a) 5 marks total

0 marks if unattempted.

2(3) marks if a student tried to explain using plain english, but no numerical example given.

2 marks if an example is designed, but the example is incorrect.

5 marks if correct demonstration of a counter example.

-0.5/-1 for missing m depending on the severity of the mistake

part(b) 5 marks total

0 marks if unattempted.

2 marks if a student tried to explain using plain english, but no numerical example given.

2 marks if an example is designed, but the example is incorrect.

5 marks if correct demonstration of a counter example.

-0.5/-1 for missing m depending on the severity of the mistake

part(c) 8 marks total

0 marks if unattempted.

3 marks if any kind of sorting is mentioned by the student, but the rest of the logic is incorrect.

4 marks if events are arranged in decreasing order of funds, but the rest of the algorithm is incorrect.

4 marks for iteratively assigning events on the last day available or before its deadline, but the order of events is incorrect.

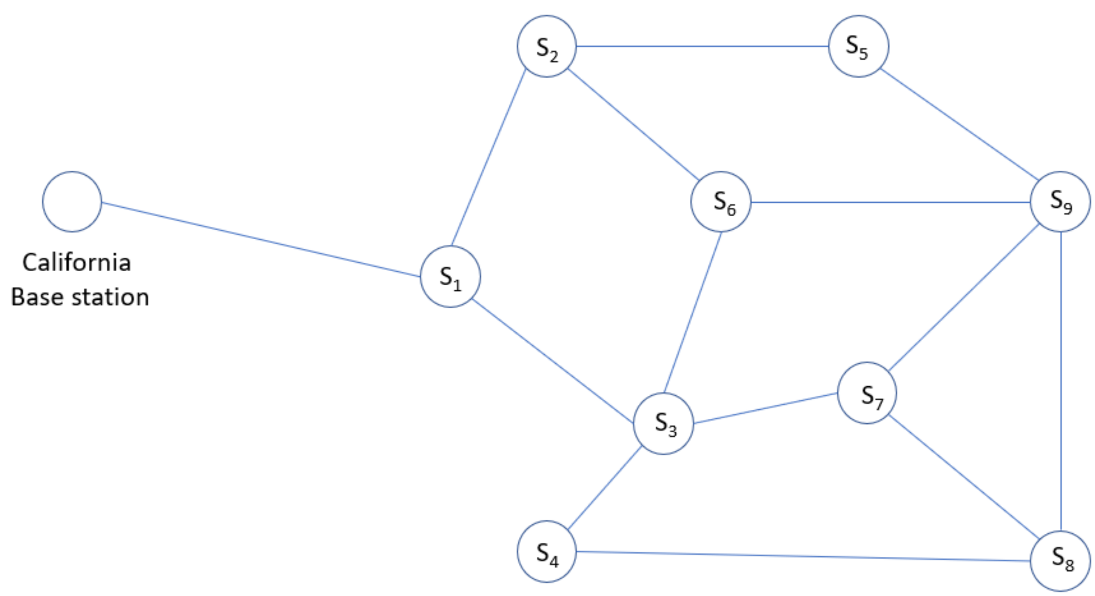
8 marks for a correct algorithm.

-2 if not mentioned the event is skipped if no day is available.

6. 16 pts

Because of a bad update, you (the CTO of SpaceX) are in a hurry to fix the ongoing issue with the Starlink satellite. Your talented engineers have prepared the urgent fix, which needs to be sent to all the satellites as soon as possible.

The fix is first sent from the base station situated in California to satellite S_1. From there, you can send the fix from satellite S_i to a satellite S_j if there is a link (edge) between S_i and S_j. You know the time required to send the fix over this link from satellite S_i to satellite S_j is T_{ij}. We can assume that if a satellite S_i needs to send the fix to k of its neighbors the k messages can be sent out at the same time. In other words, messages from a satellite to its neighbors go out in parallel not sequentially. However, a satellite can send the fix to its neighbors only AFTER it has fully received it from a neighbor.



Note: The above diagram is only an example. The network may have any number of satellites in it.

a. Provide an algorithm to determine the minimum time required to propagate the update to all satellites. In other words, we need to minimize the time between the first message leaving the base station and when the last satellite completes receiving the update (8 pts)

b. Assuming that each satellite requires a single processing time D_i before broadcasting the fix to any nearby satellites that need to receive the fix, how would you modify the solution in part a to determine the minimum time required to propagate the fix to all satellites. (8 pts)

a) Run Dijkstra's on the undirected graph and find the shortest distance to the furthest node from the California base station.

4 marks: adopting Dijkstra's algorithm

4 marks: how to apply Dijkstra's algorithm to compute the total time

b) Turn the satellite network into a directed graph G (one undirected edge -> two directed edges in opposite directions). For every (S_i, S_j) in E, let the edge weight be E_(S_i, S_j) = T_ij + D_i. Run Dijkstra's on this directed graph and find the shortest distance to the furthest node from the California base station.

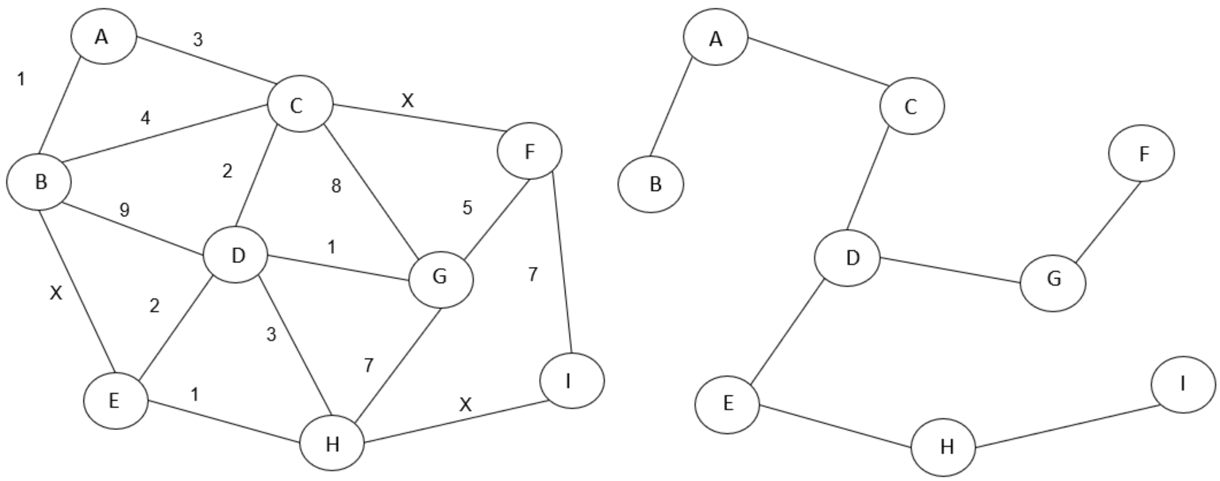
4 marks: correct graph modification, from undirected to undirected

4 marks: correct weight in the new graph: original travel time T plus processing time D

b) Alternate solution: Model each satellite i as a directed edge (S_i1, S_i2) with cost D_i. Then turn the satellite network into a directed graph G where for an undirected edge (Si, Sj) we will have edges (S_i2, Sj1) and (S_j2, S_i1). Run Dijkstra's on this directed graph and find the shortest distance to the furthest node from the California base station.

7. 13 pts

Consider the weighted undirected graph *G* on the left (see below graphs). Suppose the graph *T* on the right is the unique MST of *G*. Find the value of X, assuming X is an integer. You must provide the reasoning for your answer.



Answer:

1. If X < 5 then FG will not be the lowest connection cost between F and the rest of the MST. So, X ≥ 5. But if X=5 then the MST will not be unique since FC can replace FG without affecting the cost of the MST. So it must be that X > 5.
2. If X > 7 then FI will be the lowest connection cost between I and the rest of the MST, the MST will include the edge FI instead of HI. So, X ≤ 7. But if X=7 then the MST will not be unique since FI can replace HI without affecting the cost of the MST. So it must be that X < 7.
3. Since edge costs are integer and X < 7 and X > 5 then X=6.
4. Arguments that use a specific MST algorithm are also acceptable.

Rubrics:

1. 3 pts for the exact final answer, -1 if only a reasonable range is given
2. 10 points for the correct reasoning:
   - 5 points for the lower bound computation (X > 5), -2 points if the reasoning is incomplete (eg if the bound is not tight or the uniqueness is not mentioned or the argument is insufficient)
   - 5 points for the upper bound (X < 7), -2 points if the reasoning is incomplete (eg if the bound is not tight or the uniqueness is not used or the argument is insufficient)

8. 10 pts

Consider the following set of preference lists for three men and women:

m_1	m_2	m_3	w_1	w_2	w_3
w_2	w_1	w_3	m_2	m_3	m_1
w_1	w_2	w_1	m_3	m_1	m_2
w_3	w_3	w_2	m_1	m_2	m_3

True or false: m_1 is a valid partner of w_1. If true, give a stable matching in which m_1 and w_1 are matched; if false, prove that m_1 is not a valid partner of w_1.

Answer: False

Brief Explanation: Suppose m_1 is a valid partner of w_1, i.e. there exists a stable matching *S* in which m_1 and w_1 are matched. Then either m_2 is matched with w_2 and m_3 is matched with w_3 in *S*, or m_2 is matched with w_3 and m_3 is matched with w_2 in *S*. If m_2 is matched with w_2 and m_3 is matched with w_3 in *S*, then (m_1, w_2) is an instability in *S*, since m_1 prefers w_2 over their current partner w_1 and w_2 prefers m_1 over their current partner m_2. If m_2 is matched with w_3 and m_3 is matched with w_2, then (m_3, w_1) is an instability, since m_3 prefers w_1 over their current partner w_2 and w_1 prefers m_3 over their current partner m_1. In either case, *S* is not a stable matching, a contradiction. Therefore, for the given preference lists, there does not exist a stable matching in which m_1 is matched with w_1, and so m_1 is not a valid partner of w_1.

Rubrics:

0pts. if blank or only marked true

2pts. for marking true and if some explanation is provided (eg. if some student noted down wrong preference order and in that ordering, stable matching exists)

4pts. if marked false but no explanation provided

5pts. if marked false and explained instability only through GS algorithm's iteration or through 1 matching instance (incomplete proof - it only proves that m1 and w1 are not a valid pair with the GS-algorithm).

Explanation: Analyzing the behavior of the GS algorithm is not enough. Consider the following example: the preference list of 3 men and women are given as, m1 = {w1,w2,w3}, m2 = {w2,w3,w1}, m3 = {w3,w1,w2}, w1 = {m2,m3,m1}, w2 = {m3,m1,m2}, w3 = {m1,m2,m3}. Clearly, when men propose first, the computed matching would be {(m1,w1),(m2,w2),(m3,w3)}; when women propose first, the computed matching would be {(w1,m2),(w2,m3),(w3,m1)}. However, {(m1,w2),(m2,w3),(m3,w1)} is also a stable matching, even though none of its pairs appears in the two stable matchings computed by the GS algorithm.

10pts. if marked false and mentioned that m2 and w1 prefer each other the most and will hence always end up together, without explaining instabilities. The fact that m2 and w1 prefer each other the most will automatically ensure that (m2,w1) in the stable matching, which is proved in Problem 2 of Homework 1, Kleinberg and Tardos, Chapter 1, Exercise 2.

10pts. if marked false and mentioned the GS results and the following fact: in the computed stable matching S, each woman is paired with her worst valid partner (Lemma (1.8) in Section 1.1, Kleinberg and Tardos). Similarly, if mentioned (m2,w1) appears in both the computed stable matchings of men proposing and women proposing, the student can get full score.

10pts. if marked false and given complete proof and discussed thoroughly.

(*) -2 pts for each incorrect statement in the solution's explanation.