CSE 332 : 22Su Final Pt. 2 Solutions

Name:	**NetID:** @uw.edu

Instructions

- The allotted time is **60** minutes. Please do not turn the page until the staff says so.
- This is a closed-book and closed-notes exam. You are not permitted to access electronic devices.
- Read directions carefully, especially for problems that require you to show work or provide an explanation.
- We can only give partial credit for work that you've written down.
- Unless otherwise noted, every time we ask for an O, Ω, or Θ bound, it must be simplified and tight.
- For answers that involve bubbling ◯ or ☐, make sure to fill in the shape completely: ● or ■.
- If you run out of room on a page, indicate that the answer continues on the back of that page. Try to avoid writing on the very edges of the pages: we scan your exams and edges often get cropped off.
- Make sure you also get a copy of the formula sheet.

Advice

- If you feel like you're stuck on a problem, you may want to skip it and come back at the end if you have time.
- Look at the question titles on the cover page to see if you want to start somewhere other than problem 1.
- Remember to take deep breaths.

Question	**Max points**
1. Quick Facts	20
2. Hashing	8
3. MST	10
4. Shortest Paths	12
5. Topological Sorts	6
6. P, NP, NP-Complete	12
Total	**68**

1. Quick Facts [20 points]

(a) Given a B-tree of height 4, with `M=3` and `L=5` what is the **MINIMUM** number of data items in the tree? You can leave your answer as a multiplication, there's no need to simplify the answer. **Solution:**

$2 \cdot 2^3 \cdot 3 = 48$

(b) Given a B-tree of height 4, with `M=3` and `L=5`, what is the **MAXIMUM** number of data items in the tree? You can leave your answer as a multiplication, there's no need to simplify the answer. **Solution:**

$3^4 \cdot 5 = 405$

(c) Consider a hashtable that uses separate chaining as its collision resolution strategy and has a load factor of 332 and n elements. What is the **WORST CASE** runtime of a find operation in this hashtable? **Solution:**

$O(n)$

(d) True or false: if a hastable uses quadratic probing and its table size is prime, it is always guaranteed to find an empty slot. **Solution:**

False, it also has to have a load factor < 0.5

(e) True or false: consider a hashtable that uses double hashing strategy, if g(key) and h(key) is the same function, the insert operation will **ALWAYS** result in an infinite cycle when load factor is greater than 0.5. **Solution:**

False, might happen but not always.

(f) **BEST CASE** runtime for insertion sort of n elements. **Solution:**

$O(n)$

(g) What is the span of parallel partitioning n elements into three groups: elements less than pivot, elements greater than pivot, and the pivot. **Solution:**

$O(\log n)$

(h) True or false: A program can have concurrency issues without multiple processors. **Solution:**

True, a single processor can still have concurrency issues so long as there are multiple threads.

(i) True or false: One benefit using re-entrant lock is that it eliminates all possibilities of potential deadlock situations. **Solution:**

False, does not eliminate ALL possibilities.

(j) **WORST CASE** runtime of running DFS on a dense graph with `V` nodes and `E` edges. **Solution:**

$\mathcal{O}(E)$ or $\mathcal{O}(V+E)$ or $\mathcal{O}(V^2)$ (because of **dense** graph)

2. Hashing [8 points]

Two of the greatest soccer teams in the world, FC Barcelona and Real Madrid, are trying to figure out how to place their players on the team bus. Their buses have 10 empty spots, and they came up with the brilliant idea to hire a CS major to write some code for assigning seats.

- FC Barcelona hires an intern from UW who suggests that FC Barcelona can use a hash function with ***linear probing*** for collision resolution.
- Real Madrid hires an intern from UC Berkeley who suggests that Real Madrid can use a hash function but use ***quadratic probing*** for collision resolution.

(a) (4 points) Use ***linear probing*** and ***quadratic probing*** respectively to insert the following players for both teams. If a value cannot be inserted, indicate this and continue inserting the remaining elements.

Use the primary hash function $h(k) = k\%10$:

$$1, 11, 21, 2, 3, 12, 5, 15, 7, 10$$

Solution:

0	10
1	1
2	11
3	21
4	2
5	3
6	12
7	5
8	15
9	7

FC Barcelona (Linear Probing)

0	15
1	1
2	11
3	2
4	3
5	21
6	12
7	7
8	
9	5

Real Madrid (Quadratic Probing)

10 cannot be inserted to quadratic probing.

(b) (2 points) Which of the two hash functions works better in this scenario and why? Answer in 1-2 sentences.
Solution:

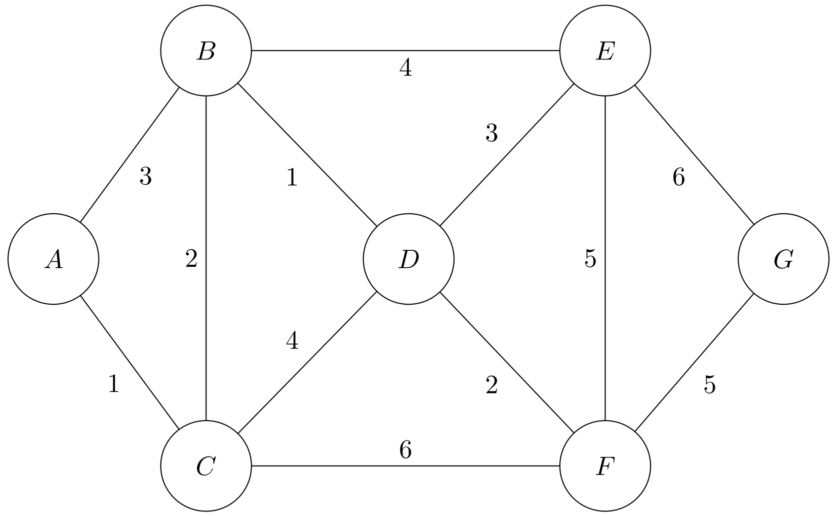
The FC Barcelona one would be better even though we need more probes. The Real Madrid one does not even completely work since there is no space for the last 10. Linear probing guarantees that every spot will get filled if there is one available.

(c) (2 points) What is ***one*** advantage and ***one*** disadvantage of separate-chaining as a collision resolution strategy?
Solution:

Many possible answers. Possible advantage include not having to have a load factor less than one. One disadvantage is it takes more space.

3. MST [10 points]

You do **NOT** need to show work for this question.



(a) (5 points) Bubble all the valid orderings of edges being added when using **Prim's algorithm**

Solution:

○ AC BD BC DE DF FG

● AC BC BD DF DE FG

○ AC BC BD DF FG DE

● FG DF BD BC AC DE

○ FG DF BC AC BD DE

(b) (5 points) Bubble all the valid orderings of edges being added when using **Kruskal's algorithm**

Solution:

● AC BD BC DF DE FG

○ AC BC DE DF FG BD

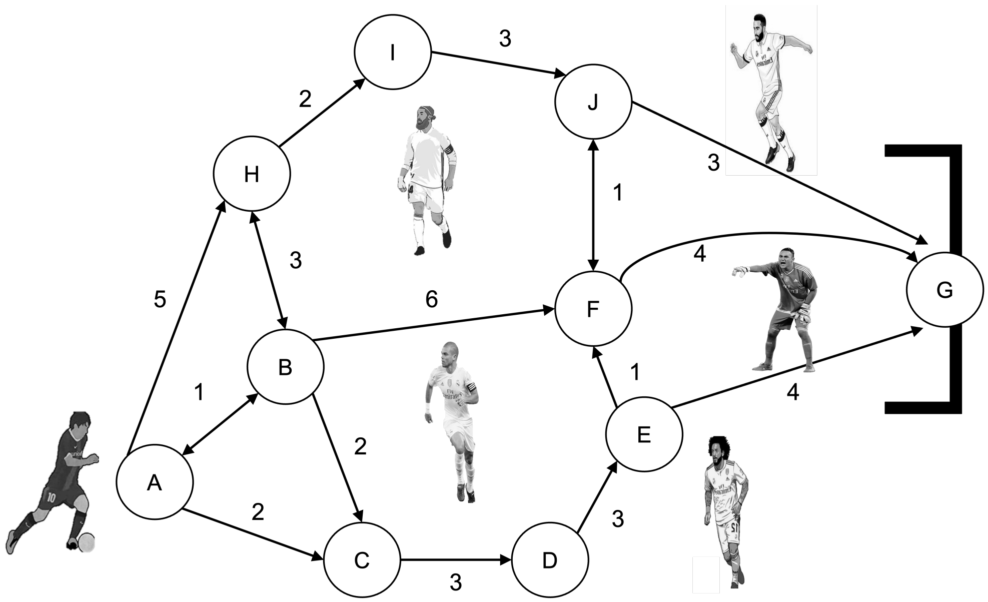
● BD AC BC DF DE FG

○ AC BD DE FG DF BC

○ BD AC BC DE DF FG

4. Shortest Paths [12 points]

The stage is set. The final of the UEFA Champions League. FC Barcelona vs Real Madrid. El Clásico. It is the last minute of the game, and the score is tied. Barcelona get the ball at the half-way line and they make one final push towards goal. The ball is at the feet of FC Barcelona's G.O.A.T, Lionel Messi, who starts at Node A on the graph below:



Messi wants to calculate the shortest path to each position on the field (each node on the graph).

(a) For full credit, you **must show all of your steps** in the table by **crossing through Distance and Path values** that are replaced by a new value. Break ties by alphabetical order; ex. if B and C were tied, you would explore B first. *Note that the next question asks you to recall what order vertices were declared processed.*

Solution:

Vertex	Processed?	Distance	Predecessor
A	Y	0	–
B	~~N~~ Y	~~∞~~ 1	A
C	~~N~~ Y	~~∞~~ 2	A
D	~~N~~ Y	~~∞~~ 5	C
E	~~N~~ Y	~~∞~~ 8	D
F	~~N~~ Y	~~∞~~ 7	B
G	~~N~~ Y	~~∞~~ 11	F
H	~~N~~ Y	~~∞~~ ~~5~~ 4	~~A~~ B
I	~~N~~ Y	~~∞~~ 6	H
J	~~N~~ Y	~~∞~~ ~~9~~ 8	~~I~~ F

(b) (1 point) In what order would Dijkstra's algorithm mark each node as processed? **Solution:**

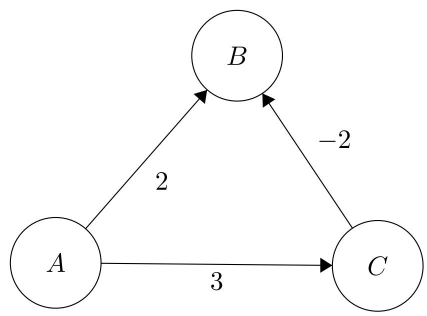
A, B, C, H, D, I, F, E, J, G

(c) (1 point) What is the shortest path (ie: nodes to visit) that Messi should take to get to the goal (ie: A to G)? **Solution:**

Shortest Path: A B F G

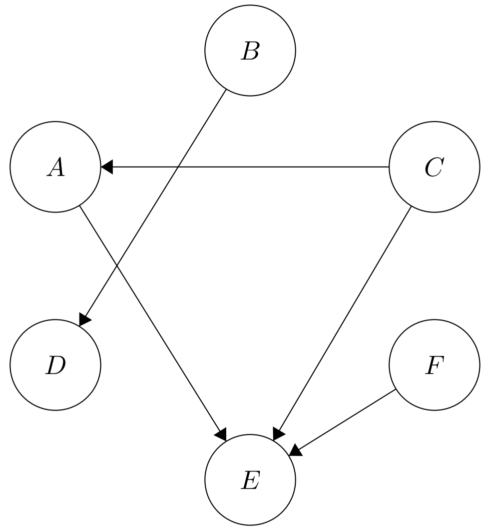
(d) (4 points) **Unrelated to previous graph!** Let's say we wanted to consider new graphs where there are negative edges, but the graph also does not have any negative weight cycles. Would we be able to use Dijkstra's algorithm in this scenario? If so, why? If not, give an example where Dijkstra's algorithm gives the wrong answer and explain. **Solution:**

No. Lots of possible examples, one possible answer:



5. Topological Sorts [6 points]

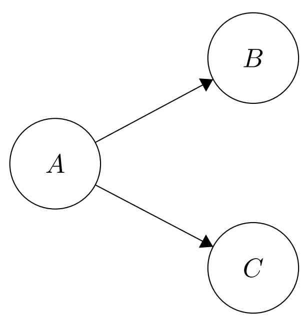
(a) (3 points) Give **two** valid topological sorts for the following graph.



Solution:

CAFEBD, BDCAFE, BFCADE, etc. There are 45 valid topological sorts for this graph.

(b) (3 points) Give an example of a graph with **exactly two** distinct topological sorts and write out both orderings.
Solution:



ABC and ACB are the only two valid toposorts for this graph.

6. P, NP, NP-Complete [12 points]

(a) (2 points) "NP" stands for Nondeterministic Polynomial

(b) (2 points) What does it mean for a problem to be in NP?
Solution:

NP is the set of all problems for which a given candidate solution can be tested in polynomial time.

(c) (2 points each) For each of the following problems, bubble all the categories which the problem is **known** to be in.

Solution:

Determining if an array is sorted.	■ P	■ NP	☐ NP-complete	☐ None of these
Determining if a chess move is the best move on an N x N board	☐ P	☐ NP	☐ NP-complete	■ None of these
Determining if there is a walk that begins and ends at the same vertex and visits every node exactly once. The walk must also have a cost of $<$ k.	☐ P	■ NP	■ NP-complete	☐ None of these
Determining if there is a topological ordering for a directed acyclic graph representing dependencies.	■ P	■ NP	☐ NP-complete	☐ None of these

Extra piece of paper for scratch work

Reference Sheet

Geometric series identities

$$\sum_{i=0}^{k} c^i = \frac{c^{k+1}-1}{c-1} \qquad \sum_{i=0}^{\infty} c^i = \frac{1}{1-c} \text{ if } |c| < 1$$

Sums of polynomials

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2} \qquad \sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \qquad \sum_{i=0}^{n} i^3 = \frac{n^2(n+1)^2}{4}$$

Log identities

$$b^{\log_b(a)} = a \qquad \log_b(x^y) = y \cdot \log_b(x) \qquad a^{\log_b(c)} = c^{\log_b(a)} \qquad \log_b(a) = \frac{\log_d(a)}{\log_d(b)}$$

Exponent properties

$$(a^m)^n = a^{m \cdot n} = (a^n)^m$$