Midterm2

Graded

Student

Jesse Zhang

Total Points

37 / 74 pts

Question 1

MC1

0 / 3 pts

– 0 pts Correct

✔ **– 3 pts** Incorrect

Question 2

MC2

0 / 3 pts

– 0 pts Correct

✔ **– 3 pts** Incorrect

Question 3

MC3

3 / 3 pts

✔ **– 0 pts** Correct

– 3 pts Incorrect

Question 4

MC4 **0** / 9 pts

4.1 MC4a **0** / 0.5 pts

− **0 pts** Correct

✔ − **0.5 pts** Incorrect

4.2 MC4b **0** / 0.5 pts

− **0 pts** Correct

✔ − **0.5 pts** Incorrect

4.3 MC4c **0** / 0.5 pts

− **0 pts** Correct

✔ − **0.5 pts** Incorrect

4.4 MC4d **0** / 0.5 pts

− **0 pts** Correct

✔ − **0.5 pts** Incorrect

4.5 MC4e **0** / 0.5 pts

− **0 pts** Correct

✔ − **0.5 pts** Incorrect

4.6 MC4f **0** / 0.5 pts

− **0 pts** Correct

✔ − **0.5 pts** Incorrect

4.7 MC4g **0** / 0.5 pts

− **0 pts** Correct

✔ − **0.5 pts** Incorrect

4.8 MC4h **0** / 0.5 pts

− **0 pts** Correct

✔ − **0.5 pts** Incorrect

4.9 MC4i **0** / 0.5 pts

− **0 pts** Correct

✔ − **0.5 pts** Incorrect

4.10 MC4j — 0 / 0.5 pts

- 0 pts Correct

✔ - 0.5 pts Incorrect

4.11 MC4k — 0 / 0.5 pts

- 0 pts Correct

✔ - 0.5 pts Incorrect

4.12 MC4l — 0 / 0.5 pts

- 0 pts Correct

✔ - 0.5 pts Incorrect

4.13 MC4m — 0 / 0.5 pts

- 0 pts Correct

✔ - 0.5 pts Incorrect

4.14 MC4n — 0 / 0.5 pts

- 0 pts Correct

✔ - 0.5 pts Incorrect

4.15 MC4o — 0 / 0.5 pts

- 0 pts Correct

✔ - 0.5 pts Incorrect

4.16 MC4p — 0 / 0.5 pts

- 0 pts Correct

✔ - 0.5 pts Incorrect

4.17 MC4q — 0 / 0.5 pts

- 0 pts Correct

✔ - 0.5 pts Incorrect

4.18 MC4r — 0 / 0.5 pts

- 0 pts Correct

✔ - 0.5 pts Incorrect

Question 5

MC5

3 / 3 pts

- ✔ **− 0 pts** Correct

− 3 pts Incorrect

Question 6

MC6

3 / 3 pts

- ✔ **− 0 pts** Correct

− 3 pts Incorrect

Question 7

True/False **16** / 20 pts

7.1 TF1 **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect

7.2 TF2 **0** / 2 pts

– 0 pts Correct

✔ **– 2 pts** Incorrect

7.3 TF3 **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect

7.4 TF4 **0** / 2 pts

– 0 pts Correct

✔ **– 2 pts** Incorrect

7.5 TF5 **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect

7.6 TF6 **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect

7.7 TF7 **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect

7.8 TF8 **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect

7.9 TF9 **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect

7.10 **TF10** **2** / 2 pts

- ✔ **− 0 pts** Correct
- **− 2 pts** Incorrect

Question 8

Algorithm execution **2** / 14 pts

8.1 **(a)** **2** / 10 pts

- **− 0 pts** Correct
- **− 10 pts** No answer
- **− 10 pts** Completely incorrect
- ✔ **− 6 pts** Only first round correct
- **− 4 pts** Only first two rounds correct
- **− 2 pts** Final answer incorrect
- ✔ **− 2 pts** Multiple minor mistakes
- **− 1 pt** Minor mistake

8.2 **(b)** **0** / 4 pts

- **− 0 pts** Correct
- ✔ **− 4 pts** No answer
- **− 3 pts** Wrong answer about the existence of negative cycle
- **− 2 pts** Correct answer about the existence of negative cycle but didn't draw the cycle/shortest-path tree
- **− 1 pt** Minor issue on the cycle/shortest-path tree

Question 9

Algorithm design **10** / 16 pts

9.1 (a) **8** / 8 pts

✔ **– 0 pts** Correct

– 8 pts No answer

– 4 pts Completely incorrect path

– 2 pts Correct path but incorrect total time

9.2 (b) 2 / 8 pts

– 0 pts Correct

– 8 pts No answer

Algorithm

– 0 pts Completely correct

– 5 pts Completely incorrect algorithm

– 4 pts Algorithm is hard-coded or only works in specific cases

✔ **– 3 pts** Has the right idea but requires significant revision or correction.

– 2 pts Has the right idea and can be made correct with some effort

– 1 pt Not fully correct but the mistake can be easily fixed

– 0.5 pts other minor mistakes

Correctness

– 0 pts Completely correct

✔ **– 2 pts** No correctness argument

– 1 pt Argument has major flaws

– 0.5 pts Has the right idea but the argument is not fully correct

---

Running time

– 0 pts Match the requirement

– 0.5 pts Doesn't match the requirement but is acceptable

✔ **– 1 pt** Extremely inefficient (e.g., brute force)

Pennsylvania State University — CMPSC465 : Data Structures & Algorithms
Lecturers: Ke Chen and Yana Safonova

Exam 2
October 17, 2025

Midterm 2 - D

Name: Jesse Zhang

Penn State access ID (xyz1234) in the following box:

zfz5311

Instructions:

- Answer all questions. Read them carefully first. Be precise and concise. Handwriting needs to be neat. Box numerical final answers.
- Please clearly write your name and your PSU access ID (i.e., xyz1234) in the box on top of **every page**.
- Write in only the space provided. You may use the back of the pages only as scratch paper. **Do not write your solutions in the back of pages!**
- **Do not write outside of the black bounding box**: the scanner will not be able to read anything outside of this box.
- We are providing two extra page at the end if you need extra space. Make sure you mention in the space provided that your answer continues there.

Good luck!

Name: Jesse Zhang | PSU Access ID (xyz1234): zfz5311

Multiple choice questions *(24 points)*

For each of the following questions, select the right answer by filling the corresponding grading bubble.

1. Run BuildHeap on the array $[1, 10, 2, 9, 3, 8, 4, 7, 5, 6]$ to construct a **min-heap**. Select the resulting max-heap.

   1,3,2,9,10,8,4,7,5,6
   1,3,2,5,10,8,4,7,9,6

   - ○ $[1, 2, 3, 4, 5, 9, 7, 10, 6, 8]$
   - ○ $[1, 3, 2, 5, 6, 8, 4, 7, 9, 10]$
   - ○ $[1, 3, 2, 5, 6, 8, 4, 10, 7, 9]$
   - ○ $[1, 2, 3, 4, 5, 10, 9, 8, 7, 6]$
   - ○ $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$
   - ● None of the above

2. Consider the **max-heap** $[8, 7, 6, 4, 3, 2, 5, 1]$. Which of the following is the resulting heap after inserting the number 9?

   8,7,6,4,3,2,5,1,9
   8,7,6,1,3,2,5,4,9
   8,7,2,1,3,6,5,4,9

   9,8,7,6,4,3,2,5,1
   9,8,7,1,4,3,2,5,6
   9,8,2,1,4,3,7,5,6
   9,1,2,8,4,3,7,5,6

   - ○ $[9, 8, 6, 7, 5, 4, 3, 1, 2]$
   - ○ $[9, 8, 6, 7, 4, 3, 2, 5, 1]$
   - ○ $[9, 8, 6, 7, 3, 2, 5, 1, 4]$
   - ○ $[9, 7, 8, 6, 4, 3, 2, 5, 1]$
   - ○ $[9, 8, 7, 6, 5, 4, 3, 2, 1]$
   - ● None of the above

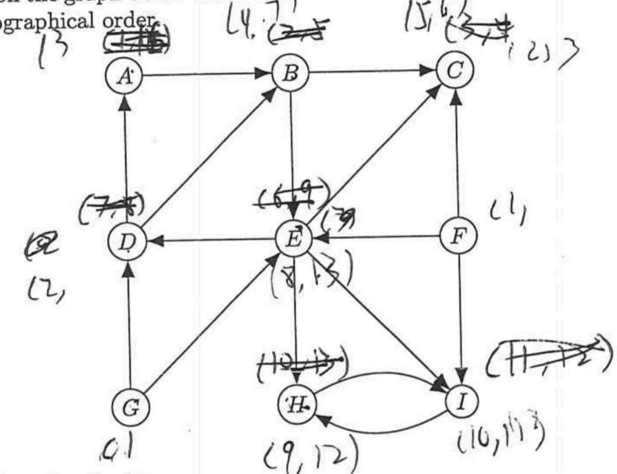
3. The array $[12, 6, 9, 5, 4, 3, 1]$ is a **max-heap**, what does the array look like after the first two iterations of the HeapSort algorithm?

   6,12,9,5,4,3,1
   6,5,9,12,4,3,1
   6,5,3,12,4,9,1 12
   6,5,3,1,4,9,12

   - ○ $[1, 3, 4, 5, 6, 9, 12]$
   - ○ $[1, 3, 12, 6, 9, 5, 4]$
   - ○ $[6, 5, 4, 1, 3, 9, 12]$
   - ● $[6, 5, 3, 1, 4, 9, 12]$
   - ○ $[6, 5, 4, 3, 1, 9, 12]$
   - ○ None of the above

Name: Jesse Zhang	PSU Access ID (xyz1234): zf5311

4. Do a depth-first search on the graph below and answer the following questions. Process vertices and edges out of a vertex in lexicographical order.



(a) The pre-visit number of node A is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(b) The pre-visit number of node B is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(c) The pre-visit number of node C is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(d) The pre-visit number of node D is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(e) The pre-visit number of node E is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(f) The pre-visit number of node F is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(g) The pre-visit number of node G is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(h) The pre-visit number of node H is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(i) The pre-visit number of node I is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(j) The post-visit number of node A is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(k) The post-visit number of node B is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(l) The post-visit number of node C is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(m) The post-visit number of node D is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

Name: Jesse Zhang **PSU Access ID (xyz1234):** zfz5311

(n) The post-visit number of node E is:
○1 ○2 ○3 ○4 ○5 ○6 ●7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(o) The post-visit number of node F is:
○1 ●2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(p) The post-visit number of node G is:
○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ●15 ○16 ○17 ○18

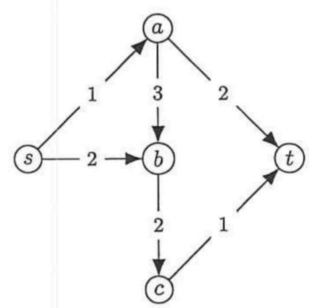
(q) The post-visit number of node H is:
○1 ○2 ○3 ○4 ○5 ●6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

(r) The post-visit number of node I is:
○1 ○2 ○3 ○4 ●5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16 ○17 ○18

5. The strongly connected components of the graph in question 4, in the order in which they would be discovered by the algorithm from class (DFS in the reversed graph in lexicographical order, then DFS in the original graph), are:

○ HI, C, F, ABDE, G
○ HI, ADGE, BCF
● HI, C, ABDE, G, F
○ HI, ABCDEFG
○ ABDE, G, HI, C, F
~~●~~ HI, ABDE, G, C, F
○ ABCDEFG,HI
○ None of the above

6. Consider the following flow network with source s and sink t. The value of the maximum flow is:



3 5 7

1/5

○ 1
○ 3/2
● 2
○ 3
○ 4

Name: Jesse Zhang	**PSU Access ID (xyz1234):** zfz5311

True/False *(20 points)*

True or false? Fill in the correct bubble. No justification is needed.

T F

● ○ Checking whether a min-heap with n elements contains a certain element takes $O(n)$ time.

● ○ In a min-heap, the largest element is stored in the last position of the array.

● ○ Multiplying each edge weight by a constant $k > 0$ will not affect the shortest paths in a graph.

● ○ The Floyd-Warshall algorithm can correctly find all-pairs shortest paths even when there are negative cycles.

● ○ In a weighted graph $G = (V, E)$ where all edge weights are positive integers and at most 465, we can find the shortest paths from a given vertex s to any other vertex in $O(|V| + |E|)$ time.

● ● In a DFS on a directed graph, the vertex with the smallest post-visit number is in a source strongly connected component.

○ ● In a DFS on a directed graph, if vertex u was explored before v, and there is a path from u to v, then $post(v)$ is greater than $post(u)$.

○ ● Every directed acyclic graph (DAG) has exactly one topological ordering.

○ ● If the weights of edges in a connected, weighted graph are all distinct, then it it not possible to have multiple shortest paths from a source to a destination.

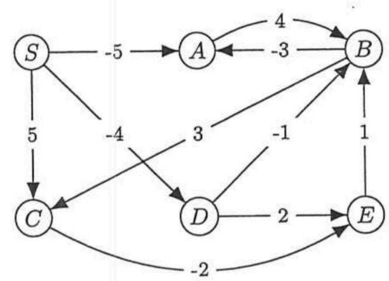
● ○ In a valid flow, for every vertex other than the source and the sink, the total incoming flow must be equal to the total outgoing flow.

Name: Jesse Zhang PSU Access ID (xyz1234): zf25311

Algorithm execution *(14 points)*

Run Bellman-Ford on the following graph to find shortest paths from node S. In each round, call Update on edges in ascending order of edge weights, namely, (S,A), (S,D), (B,A), (C,E), (D,B), (E,B), (D,E), (B,C), (A,B), (S,C).

-5+4



(a) Draw a table showing the intermediate distance values of all the nodes after each of the $|V|-1$ rounds of the algorithm.

(S,A) = -5
(S,D) = -4
(B,A) = -3
(C,E) = -2
(D,B) = -1
(E,B) = 1
(D,E) = 2
(B,C) = 3
(A,B) = 4
(S,C) = 5

(b) Is there a negative cycle in the graph? If yes, draw the negative cycle. If not, draw the final shortest-path tree.

NO ~~no~~ negative clycle

~~S,C,E~~ S, A, B, A

Yes negative cycle one

S, C, E, B, C, E then cycle

~~S, C, E, B, C~~

S A, B A = -4

S, D, B C, E, B

S D ~~A~~ E, B, C, E

Name: Jesse Zhang	PSU Access ID (xyz1234): zfz5311

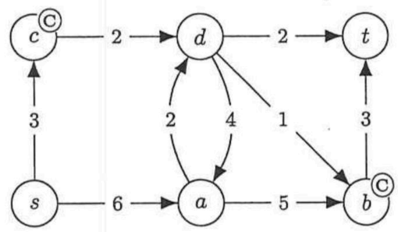
Algorithm design *(16 points)*

Consider a campus map modeled as a weighted, directed graph $G = (V, E)$, where each vertex represents a building and each edge represents a road directly connecting two buildings. Each edge $e \in E$ has a positive integer weight $t(e)$, representing the number of minutes required to travel between the two connected buildings.

Prof. Kaffee wants to walk from a starting building $s \in V$ to a target building $t \in V$. Being a caffeine enthusiast, whenever he passes a building that contains a coffee shop, he spends **an additional two minutes** enjoying an espresso. Suppose all buildings that have a coffee shop are marked with a ©, find a path from s to t that minimizes Prof. Kaffee's total travel time. (You may assume that there is no coffee shop in s and t.)

(a) In the graph below, what is the fastest path from s to t? What is Prof. Kaffee's total travel time?

7



17

fastect path: s, c, d, t   travel time 9 mins

(b) Provide an algorithm for this task and justify why it works. The running time of your algorithm should be $O((|V| + |E|) \log |V|)$.

1. Do a DFS to min-heap the Grap uth ~~isa~~ including C
2. Build DFS tree to see the most minest mins to take t ut
3. if s can go t, then it work, ~~if not mean~~
4. By the algorithm of DFS the complexity time are $O((V+E)\log V)$

Name: Jesse Zhang	**PSU Access ID (xyz1234):** zfz5311

Name: Jesse Zhang	PSU Access ID (xyz1234): zfz5311