CMPSC 465 Assignment 05

Shi Qiu

TOTAL POINTS

19 / 25

QUESTION 1

P1 5 pts

1.1 P1.1 2 / 5

✓ **+ 2 pts** proved/explained 1.a correctly.

+ 3 pts proved/explained 1.b correctly.

+ 5 pts proved/explained 1.c correctly. (Bonus)

+ 0.2 pts 10% points for part 1

+ 0.3 pts 10% points for part 2

+ 0 pts No answer/Handwritten/More than 2 hours

💬 Directed path from B to D exists, B should precede D in the linearization.

QUESTION 2

P2 10 pts

2.1 P2.1 6 / 6

✓ **+ 6 pts** Provided the correct pre and post numbers for all nodes.

+ 3 pts Provided a correct pre and post number but didn't follow alphabetical order when there is a choice of vertices.

+ 0.6 pts I don't know how to answer this question.

+ 0 pts Wrong Answer/ Handwritten/ Empty

2.2 P2.2 4 / 4

✓ **+ 4 pts** identified that metagraph is the given graph itself.(no need to draw the graph).

+ 0.4 pts I don't know how to answer this question.

+ 0 pts Wrong Answer / Unanswered

QUESTION 3

3 P3 7 / 10

+ 5 pts construct the graph correctly

+ 2 pts incorrect graph but add vertices correctly

+ 3 pts incorrect graph but add edges correctly

✓ **+ 5 pts** correct algorithm and running time analysis (correct running time can be O(|V|+|E|), O(|V|), O(|E|), or O(1) if use 400 as a constant, or O(n) if use mod n as input; using big-theta intead of big-O is also correct)

+ 1 pts incorrect algorithm but transfer the problem correctly (e.g. check if there is a path from v_1 to v_20)

+ 3 pts incorrect algorithm but apply explore algorithm

+ 1 pts incorrect algorithm but appropriate running time analysis

+ 0.5 pts incorrect algorithm but appropriate running time analysis (deduct 0.5 point if missing running time analysis for creating the graph)

+ 1 pts I don't know how to answer this question.

+ 8 pts correct graph and algorithm but no running time analysis

- 2 pts 20% penalty for any late submission in 2 hours

+ 0 pts incorrect/empty/handwritten/later than 2 hours

+ 2 Point adjustment

💬 you did not point out how many vertices do we have and how to match number to vertices but you did mention the mod algorithm on edge

Problem 2

A

source is A B

sink is J G I

B)

ACFJDHGIBE

C)

436

1.1 P1.1 **2 / 5**

✓ **+ 2 pts** proved/explained 1.a correctly.

   + 3 pts proved/explained 1.b correctly.

   + 5 pts proved/explained 1.c correctly. (Bonus)

   + 0.2 pts 10% points for part 1

   + 0.3 pts 10% points for part 2

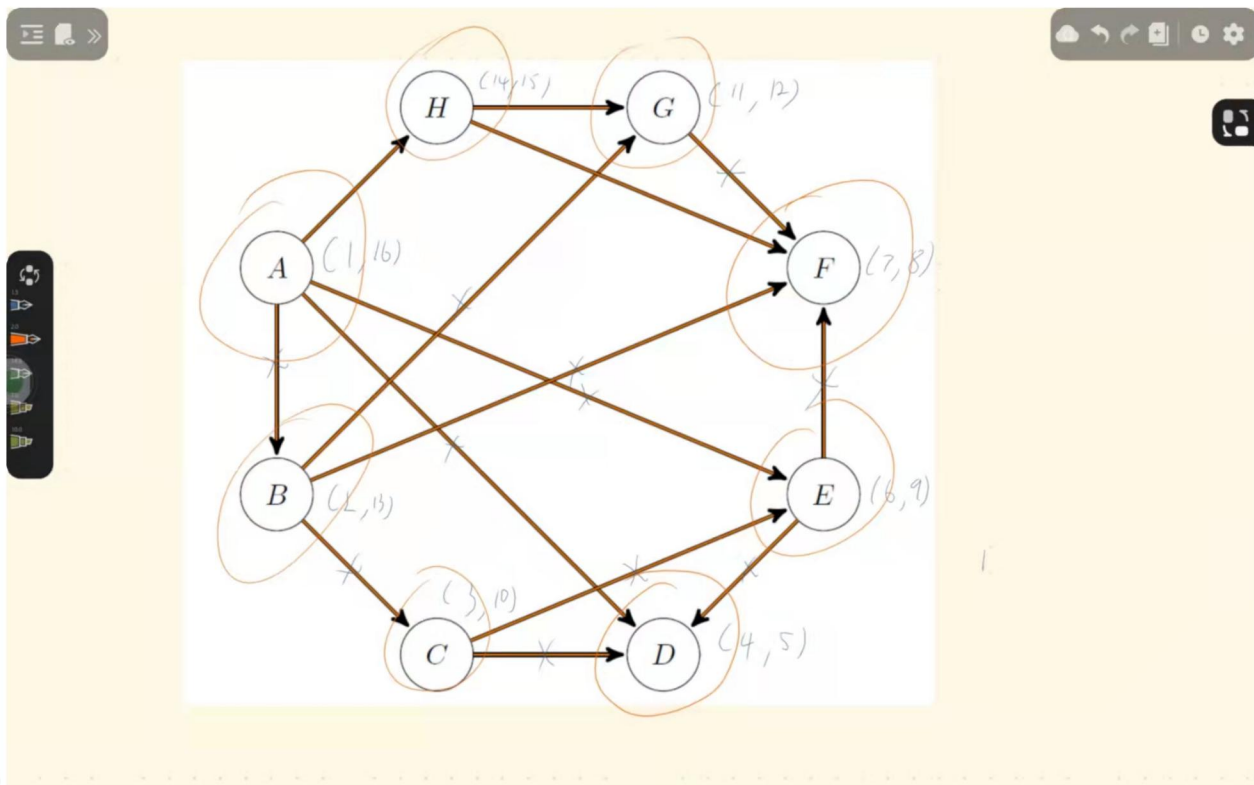   **+ 0 pts** No answer/Handwritten/More than 2 hours

   💬 Directed path from B to D exists, B should precede D in the linearization.

Problem 3

1 A(1,16) B(2,13) C(3,10) D(4,5) E(6,9) F(7,8) G(11,12) H(14,15)

2)

2.1 P2.1 6 / 6

✓ **+ 6 pts** **Provided the correct pre and post numbers for all nodes.**

+ 3 pts Provided a correct pre and post number but didn't follow alphabetical order when there is a choice of vertices.
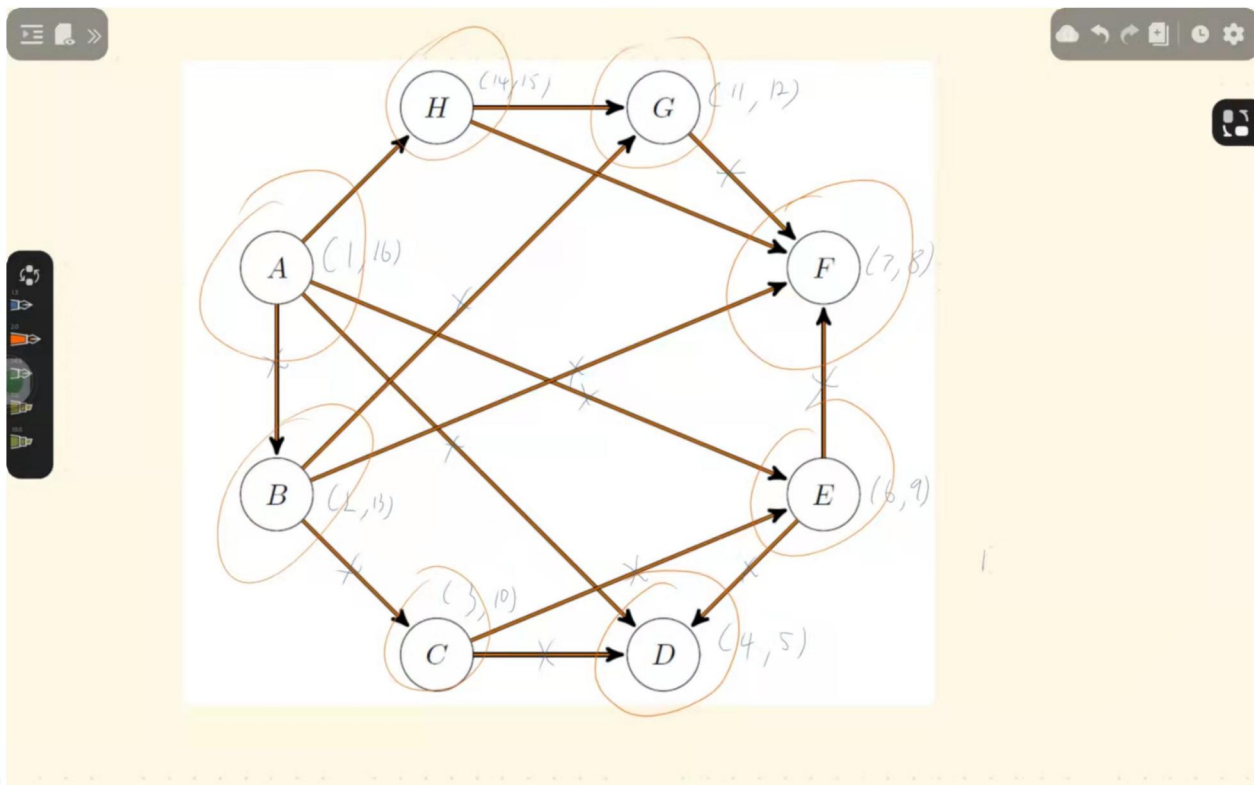
+ 0.6 pts I don't know how to answer this question.

+ 0 pts Wrong Answer/ Handwritten/ Empty

iıl gradescope

Problem 3

1 A(1,16) B(2,13) C(3,10) D(4,5) E(6,9) F(7,8) G(11,12) H(14,15)

2)

2.2 P2.2 **4 / 4**

✓ **+ 4 pts** identified that metagraph is the given graph itself.(no need to draw the graph).

+ 0.4 pts I don't know how to answer this question.

+ 0 pts Wrong Answer / Unanswered

ııl gradescope

Problem 4 | **Points:** |

if we set i dollar bill as the initial vertices, and make the second accessible dollar bill($2) as the second vertices, we can continue the graph by make every new bill we access as a new vertices and the ways of print bills as edges.

For example, the first vertices is $1, second vertices is $2, and there is a edge between them.

The third vertices will be the value of $2^2 mod 400 = 4$, connect a edge between $2 and $4, forth vertices will be $5 and connect a edge between $2 and $5

as we continues, the map keeps going until there is no more accessible bill amount. so it will be back to the first vertices: 1

and we can run depth-first search with timing, to find if there exist a bill amount is 20

the running time will be $O(v^2)$

+ **5 pts** construct the graph correctly

+ **2 pts** incorrect graph but add vertices correctly

+ **3 pts** incorrect graph but add edges correctly

✓ + **5 pts** correct algorithm and running time analysis (correct running time can be O(|V|+|E|), O(|V|), O(|E|), or O(1) if use 400 as a constant, or O(n) if use mod n as input; using big-theta intead of big-O is also correct)

+ **1 pts** incorrect algorithm

but transfer the problem correctly (e.g. check if there is a path from v_1 to v_20)

+ **3 pts** incorrect algorithm but apply explore algorithm

+ **1 pts** incorrect algorithm but appropriate running time analysis

+ **0.5 pts** incorrect algorithm but appropriate running time analysis (deduct 0.5 point if missing running time analysis for creating the graph)

+ **1 pts** I don't know how to answer this question.

+ **8 pts** correct graph and algorithm but no running time analysis

- **2 pts** 20% penalty for any late submission in 2 hours

+ **0 pts** incorrect/empty/handwritten/later than 2 hours

+ **2** Point adjustment

💬 you did not point out how many vertices do we have and how to match number to vertices but you did mention the mod algorithm on edge