Final

Graded

Student

Krishna Nitin Pagrut

Total Points

85 / 100 pts

Question 1

MCQs

45 / 50 pts

✔ **+ 2.5 pts** Q1 is correct.
Correct answer: *set 1:* **C**, *set 2:* **B**

✔ **+ 2.5 pts** Q2 is correct.
Correct answer: *set 1:* **D**, *set 2:* **A**

✔ **+ 2.5 pts** Q3 is correct.
Correct answer: *set 1:* **B**, *set 2:* **A**

✔ **+ 2.5 pts** Q4 is correct.
Correct answer: *set 1:* **A**, *set 2:* **B**

+ 2.5 pts Q5 is correct.
Correct answer: *set 1:* **B**, *set 2:* **A**

✔ **+ 2.5 pts** Q6 is correct.
Correct answer: *set 1:* **A**, *set 2:* **A**

✔ **+ 2.5 pts** Q7 is correct.
Correct answer: *set 1:* **A**, *set 2:* **D**

✔ **+ 2.5 pts** Q8 is correct.
Correct answer: *set 1:* **A**, *set 2:* **C**

✔ **+ 2.5 pts** Q9 is correct.
Correct answer: *set 1:* **B**, *set 2:* **B**

✔ **+ 2.5 pts** Q10 is correct.
Correct answer: *set 1:* **A**, *set 2:* **B**

✔ **+ 2.5 pts** Q11 is correct.
Correct answer: *set 1:* **C**, *set 2:* **C**

✔ **+ 2.5 pts** Q12 is correct.
Correct answer: *set 1:* **B**, *set 2:* **B**

✔ **+ 2.5 pts** Q13 is correct.
Correct answer: *set 1:* **B**, *set 2:* **A**

✔ **+ 2.5 pts** Q14 is correct.
Correct answer: *set 1:* **B**, *set 2:* **B**

✔ **+ 2.5 pts** Q15 is correct.
Correct answer: *set 1:* **B**, *set 2:* **B**

✔ **+ 2.5 pts** Q16 is correct.
Correct answer: *set 1:* **B**, *set 2:* **B**

✔ **+ 2.5 pts** Q17 is correct.
Correct answer: *set 1:* **B**, *set 2:* **B**

✔ **+ 2.5 pts** Q18 is correct.
Correct answer: *set 1:* **B**, *set 2:* **A**

+ 2.5 pts Q19 is correct.
Correct answer: *set 1:* **A**, *set 2:* **B**

✔ **+ 2.5 pts** Q20 is correct.
Correct answer: *set 1:* **A**, *set 2:* **A**

Question 2

DP **14** / 15 pts

+ 15 pts Everthing is correct.

✔ **+ 5 pts** Correct subproblem explanation.

+ 3 pts Partially correct subproblem explanation.

+ 0 pts Wrong/no subproblem explanation.

✔ **+ 2 pts** Correct base case.

+ 1 pt Partially correct base case

+ 0 pts Wrong/no base case

+ 5 pts Correct recurrence relation and explanation.

✔ **+ 4 pts** The recurrence relation is mostly correct but has minor errors (e.g., does not handle the edge case when i=1 and j=1, does not divide 2N-1).

+ 3 pts Significant progress has been made toward the recurrence relation.

+ 2 pts Minor progress has been made toward the recurrence relation.

+ 0 pts Wrong/no recurrence relation.

✔ **+ 3 pts** Correct time complexity analysis.

+ 1.5 pts Partially correct time complexity analysis.

+ 0 pts Wrong/no time complexity.

+ 1.5 pts I don't know.

+ 0 pts Blank

Question 3

ILP **6** / 15 pts

+ 15 pts Completely correct ILP formulation.

+ 12 pts Almost correct constraint formulation with correct objective function (OR) Completely correct constraint formulation with incorrect objective function

+ 9 pts Significant progress made to constraint relations with correct objective function

✔ **+ 6 pts** Significant progress made to constraint relations with incorrect objective function

+ 3 pts Mostly incorrect answer.

+ 1.5 pts 10% option

+ 0 pts Entirely incorrect answer/ No answer.

Question 4

Bellman-Ford **20** / 20 pts

4.1 **1.** **6** / 6 pts

- ✔ **+ 6 pts** All entries are correct (each entry in the bottom 3 rows is worth 0.5 points)
- **+ 0 pts** All entries are wrong or no answer
- **+ 0.6 pts** I do not know
- **+ 5.5 pts** 11 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 5 pts** 10 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 4.5 pts** 9 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 4 pts** 8 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 3.5 pts** 7 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 3 pts** 6 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 2.5 pts** 5 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 2 pts** 4 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 1.5 pts** 3 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 1 pt** 2 entries (out of the 12 in the bottom 3 rows) are correct
- **+ 0.5 pts** 1 entries (out of the 12 in the bottom 3 rows) are correct

4.2 **2.** **14** / 14 pts

- ✔ **+ 8 pts** Algorithm: fully correct
- **+ 5.5 pts** Algorithm: partially correct, on the right direction
- **+ 2.5 pts** Algorithm: partially correct, but with critical issue
- **+ 0 pts** Algorithm: entirely wrong or no algorithm
- **+ 0.8 pts** Algorithm: I do not know
- ✔ **+ 3 pts** Proof: fully correct
- **+ 1.5 pts** Proof: partially correct
- **+ 0 pts** Proof: entirely wrong or no proof
- **+ 0.3 pts** Proof: I do not know
- ✔ **+ 3 pts** Time Complexity Analysis: fully correct
- **+ 1.5 pts** Time Complexity Analysis: partially correct
- **+ 0 pts** Time Complexity Analysis : Wrong / No
- **+ 0.3 pts** Time Complexity Analysis: I do not know

CMPSC 465 Data Structures & Algorithms
Fall 2024 Mingfu Shao and Debarati Das

Final Exam

Time: 6:50 - 8:40PM, Monday, Dec. 16th, 2024
Set 1

Your Name: Krishna Pagrut

Your PSU Access ID: Knp5451

Your Recitation: _____

INSTRUCTIONS:

1. Please clearly write your full name, your PSU Access User ID (i.e., xyz1234), and the recitation you are in (001–010) in the box above.
2. This exam contains 23 questions.
3. For questions 1–20 (i.e., multiple-choice questions) exactly one choice is correct.
4. Your answer to questions 1–20 MUST be recorded in the table on top of page 2.
5. For questions 21–23, write your answers on the designated blank pages provided after the question.
6. You are not permitted to communicate with anyone during the exam time.
7. The exam is closed book, closed notes; no calculators, cell phones, or other electronic devices are allowed or are necessary.
8. Please make sure your handwriting is legible and use a pen to ensure the scanned copies are clear and easy to read.

Your answer to questions 1–20:

Question	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Answer	C	D	B	A	A	A	A	A	B	A	C	B	B	B	B	B	B	B	C	A

L'HEF

1. (2.5 pts.) Which of the following satisfies $f = O(g)$?

$\lim_{n\to\infty} \frac{f}{g} = \frac{2^{n\log n}}{3^n}$

$\log n!$ $\log(n!) = n\log n$

$(\log n)^2$

A. $f(n) = 2^{n\cdot\log n},\ g(n) = 3^n$

B. $f(n) = 2^n,\ g(n) = n^{\log n}$

C. $f(n) = \sum_{k=1}^{n} 1/k,\ g(n) = \sqrt{n}$

D. $f(n) = \log(n!),\ g(n) = (\log n)^2$

2. (2.5 pts.) Which of the following problems is not NP-Complete?

A. Independent Set

B. Vertex Cover

C. Hamiltonian Cycle

D. 2-SAT

3. (2.5 pts.) In a linear program, equality constraints are not allowed.

A. True

B. False

4. (2.5 pts.) Huffman tree is constructed for the following data A, B, C, D, E with frequency 0.17, 0.11, 0.24, 0.33, 0.15 respectively. Then 1000001101 is decoded as

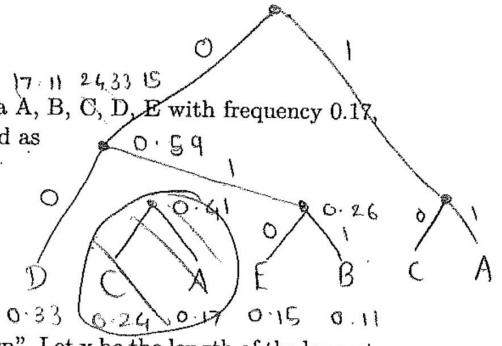
CDDA

A. BACE

B. CADE

C. BAD

D. CADD



5. (2.5 pts.) Consider two strings A = "qpqrr" and B = "pqprqrp". Let x be the length of the longest common subsequence (longest subsequence not necessarily contiguous that is present in both A and B) between A and B and let y be the number of different longest common subsequences between A and B. Then x + 10y is:

A = q p q r r

B = p q p r q r p

x = 4

y = X 2

A. 24

B. 34

C. 33

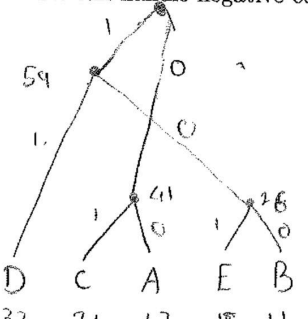
D. 44

6. (2.5 pts.) Select which of the following algorithms for MST can handle negative edge weights.

A. Both Prim's and Kruskal's

1000001101

B A C E



B. Only Prim's

C. Only Kruskal's

D. none

7. (2.5 pts.) If a problem has a linear time algorithm then it is in NP.

A. True

B. False

8. (2.5 pts.) Floyd Warshall algorithm can be applied on

A. directed weighted graphs with negative weights but not negative cycles.

B. directed weighted graphs with negative weight cycles.

9. (2.5 pts.) Let X and Y be two problems and Y can be reduced to X in polynomial time. If Y is in NP and X is NPC then Y is NPC.

A. True

B. False

10. (2.5 pts.) Any optimal solution to the dual linear program is related to the optimal solution of the primal linear program via

A. equality

B. being an upper bound

C. being a lower bound

D. All of the above

E. not related

11. (2.5 pts.) In the max flow algorithm, an augmenting path in a residual graph always has

A. only forward edges

B. only backward edges

C. depends on the current flow value associate with the residual graph

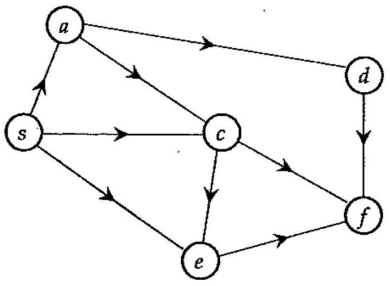
D. both forward and backward edges

12. (2.5 pts.) In dynamic program the subproblems are always disjoint.

A. True

B. False

13. (2.5 pts.) How many linearizations does the following directed acyclic graph (DAG) have?



s a c e f x 3

A. 2
B. 3
C. 4
D. 5

14. (2.5 pts.) In a dynamic programming algorithm if the size of the DP table is $M \times N$ then the runtime of the algorithm is $O(MN)$.

A. True
B. False

15. (2.5 pts.) Consider running DFS-with-timing on a directed graph $G = (V, E)$. Assume that u is explored before v. Assume also that u can reach v and v can reach u in G. Which one of the following is always true?

A. $pre[u] < pre[v]$ and $post[u] < post[v]$.
B. $pre[u] < pre[v]$ and $post[u] > post[v]$.
C. $pre[u] > pre[v]$ and $post[u] < post[v]$.
D. $pre[u] > pre[v]$ and $post[u] > post[v]$.
E. None of the above is always true.

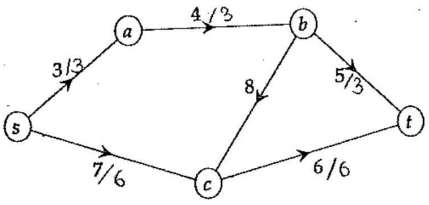
pre(u)
[[ ]]
pre(v) post(v) post(u)

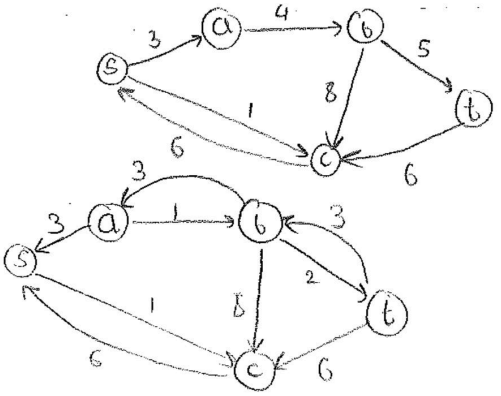
u → v

16. (2.5 pts.) Select the correct statement.

A. Any prefix-free code is a Huffman code
B. Any Huffman code is a prefix-free code
C. Both A and B
D. None of the above

17. (2.5 pts.) What is the value of the maximum flow of the network (source s, sink t) given below?





A. 8.

✓~~B~~. 9.

C. 10.

D. 11.

18. (2.5 pts.) Let X and Y be two problems and Y can be reduced to X in polynomial time. Which of the following is false? $Y \leq_P X$

A. If X can be solved in polynomial time then Y can be solved in polynomial time.

✓~~B~~. If X can not be solved in polynomial time then Y cannot be solved in polynomial time.

C. If Y can not be solved in polynomial time then X cannot be solved in polynomial time.

19. (2.5 pts.) Consider the two recursive functions: $f(n) = 2f(n/2)+n^2$, $g(n) = 4g(n/2)+n$. Which of the following is correct regarding the growth rates of $f(n)$ and $g(n)$?

$f \Rightarrow 2>1 \quad O(n^2 \log n)$

$g \Rightarrow 1<2 \quad O(n^2)$

$\frac{f}{g} = \log n \to \infty$ as $n \to \infty$

A. $f = \Theta(g)$

B. $f = O(g)$ but $f \neq \Theta(g)$

✓~~C~~. $f = \Omega(g)$ but $f \neq \Theta(g)$

20. (2.5 pts.) Let C be one minimum s-t cut of a network. Let $e \in C$ be one edge in it and we assume its capacity $c(e) \geq 2$. Suppose that we decrease the capacity of e by 1, then in the resulting network the value of the maximum flow will decrease by 1.

✓~~A~~. True for all networks.

B. True for some networks but not all.

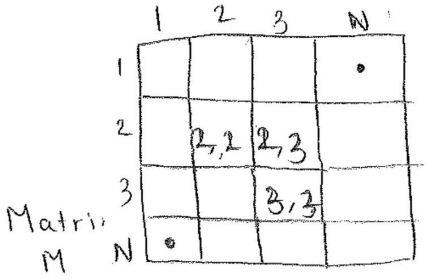
C. False for all networks.

C = min s-t cut

$e \in C \qquad c(e) \geq 2$

e is saturated

21. (15 pts.) Given a square matrix of size $N \times N$ where each cell is associated with a specific cost. A path is defined as a specific sequence of cells that starts from the bottom-left cell and move only right or up and ends on top right cell. We want to find a path with the maximum average cost over all existing paths. The average is computed as the total cost divided by the number of cells visited in the path. Write a dynamic program algorithm for this. Analyze the running time.



Let M be our $N \times N$ sq. matrix and $\forall i,j \in [1, N]$, $M[i][j] = m_{ij}$ be the cost of a cell.

$i, j-1 \rightarrow i, j$
$\uparrow$
$i+1, j$

Algorithm =>

Subproblem Def: $dp(i,j)$ is the max average cost to move from bottom left to cell (i,j)

Case 1: cell (i,j) is entered from "right" movement i.e from $(i,j-1)$

$$dp(i,j) = \frac{\left(dp(i,j-1) \cdot (N-i+(j-1)) + m_{ij}\right)}{(N-i)+j}$$

Case 2: cell (i,j) is entered from "top" movement i.e from $(i+1,j)$

$$dp(i,j) = \frac{\left(dp(i+1,j) \cdot (N-(i+1)+j) + m_{ij}\right)}{(N-i)+j}$$

Final problem to solve: $dp(1,N)$

Recurrence relation:

$$dp(i,j) = \left(\max\{dp(i,j-1) \cdot (N-i+(j-1)), dp(i+1,j) \cdot (N-(i+1)+j)\} + m_{ij}\right) / (N-i+j)$$

Base Case: $dp(N,0) = m_{N1} = M[N][1]$

$\forall i \in [1,N]$, $dp[i,1] = (dp(i-1,1) \cdot (N-(i-1)) + m_{i1})/(N-i)$

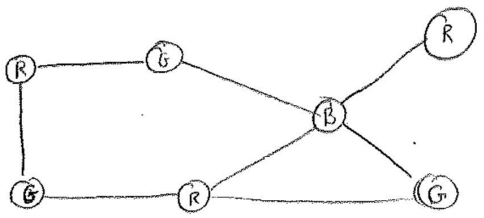
$\forall j \in [1,N]$, $dp(N,j) = (dp(N,j-1) \cdot (j-1) + m_{Nj})/j$

Run Time Analysis ⇒

There are N^2 elements in our DP matrix each requing $O(1)$ arithmatic calc time.

So runtime is $O(N^2)$

22. (15 pts.) In the graph coloring problem, we are given an undirected, unweighted graph $G = (V, E)$. The objective is to find the minimum number of colors required to color each vertex so that no two adjacent vertices get the same color. Formulate this problem using an integer linear program (ILP).



This problem is similar to minimum Vertex Cover.

So, in $G = (V, E)$ $\forall (u,v) \in E$ colour $(u) \neq$ colour (v)

Variables ⇒ Lets Lable vertices in graph as $v_i \in V$, $i \in \{1 \cdots n\}$

Indicator var ⇒ $x_{ij} = \begin{cases} 1, & (v_i, v_j) \in E \\ 0, & (v_i, v_j) \notin E \end{cases}$; $v_i, v_j \in V$

$y_{ij} = \begin{cases} 1, & v_i, v_j \in V \text{ have diff colour} \\ 0, & v_i, v_j \in V \text{ have same colour} \end{cases}$

Obj fun ⇒ $\min \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}\, y_{ij}$

When $(v_i, v_j) \in E$ edge exists then $y_{ij} = 1$ i.e they must have diffent colours. If not then they may or may not be same colour

$x_{ij} \leq y_{ij} \quad \forall i, j \in \{1, \cdots n\}$

$x_{ij}, y_{ij} \in \{0, 1\}$, so, $x_{ij}, y_{ij} \geq 0$

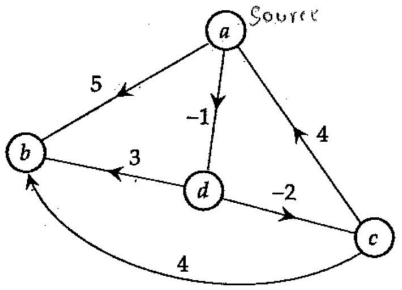
ILP ⇒ $\min \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}\, y_{ij}$

$x_{ij} \leq y_{ij} \quad \forall i, j \in \{1 \cdots n\}$

$x_{ij}, y_{ij} \in \{0, 1\} \quad \forall i, j \in \{1 \cdots n\}$

23. (6+14 pts.)

1. Run the Bellman-Ford algorithm on the directed graph given below, starting at the given source vertex a. Whenever there is a choice of edges to update, always pick the one that is lexicographically/alphabetically first. Show the dist array after each round of updates.



2. You are given a directed graph $G = (V, E)$, in which each edge $e \in E$ is assigned a positive weight $w(e)$. For any cycle C in the graph, we define the score of C as $\Pi_{e \in C} w(e)$, i.e., the product of the weights of the edges in C. You are also given a source vertex $s \in V$. Design an algorithm to determine whether in G there exists a cycle reachable from s such that its score is larger than 1. Describe your algorithm (8 pts), briefly explain its correctness (3 pts), and analyze its time complexity (3 pts).

(A1)

Edges order ⇒ ab ad ca cb db dc

Iterations ⇒ K = 1 to 3 (R_K)

iteration	a	b	c	d
R_0	0	∞	∞	∞
R_1	0	2	-3	-1
R_2	0	1	-3	-1
R_3	0	1	-3	-1

dist array finally ⇒ 0, 1, -3, -1

(A2) $G = (V, E)$ $\forall e \in E$, $w(e) > 0$

$\text{Score}(C) = \prod_{e \in C} w(e)$

Algorithm ⇒ Make a new graph $G' = (V, E)$ where

$\forall e \in E$, $l(e) = -\log(w(e))$ is length of edge 'e'

So now we have a graph $G' = (V, E)$ with neg wts and a source s

We can run Bellman-Ford negative cycle Detection. to locate a cycle with neg. wt.

Run Bellman Ford over all edges $|V|-1$ times and then run one more loop over all edges checking is dist[v] can drop even lower then we have found a cycle C with score > 1 else not.

```
def Bellman-ford_score(G, s, ∀e w(e)>0):
    Make G'
    Run Bellman-Ford |V|-1 times
    for all (u,v) ∈ E:
        if dist[v] > dist[u] + l(u,v):
            return "Found"
    return "Not Found"
```

Correctness ⇒ Proof:

We have to check the boolean stm $\prod_{e \in c} w(e) > 1$

To convert it to sum we take log( ) on both sides.

$$\prod_{e \in c} w(e) > 1 \equiv \log\left(\prod_{e \in c} w(e)\right) > \log(1)$$

[Log (AB) = Log(A) + Log(B)]

$$\Rightarrow \sum_{e \in c} \log(w(e)) > 0 \equiv -\sum_{e \in c} \log(w(e)) < 0$$

$$\Rightarrow \sum_{e \in c} -\log(w(e)) < 0 \Rightarrow \sum_{e \in c} l(e) < 0$$

Thus we can see a one to one correspondence b/w

score (c) > 1 and $\sum_{e \in c} l(e) < 0$.

Since, Bellman-Ford is a correct algo, by one-to-one correspondence our algo is correct

□

Run Time Analysis ⇒

Making the new graph by changing edges Takes $O(|E|)$ and Bellman-Ford runs in $O(|V| \cdot |E|)$ for neg cycle Detection

So net runtime is $\boxed{O(|V| \cdot |E|)}$