CS101 Algorithms and Data Structures
Fall 2023
Final Exam

Instructors: Dengji Zhao, Yuyao Zhang, Xin Liu, Hao Geng

Time: Jan 22nd 10:30-12:30

INSTRUCTIONS

Please read and follow the following instructions:

- You have 120 minutes to answer the questions.
- You are not allowed to bring any papers, books or electronic devices including regular calculators.
- You are not allowed to discuss or share anything with others during the exam.
- You should write the answer to every problem in the dedicated box **clearly**.
- You should write **your name and your student ID** as indicated on the top of **each page** of the exam sheet.

Name	
Student ID	
Exam Classroom Number	
Seat Number	
All the work on this exam is my own. **(please copy this and sign)**	

This page is intended to be blank.

1. (20 points) True or False

Each of the following statements is true (T) or false (F). Write your answers in the **answer sheet**.

(a) (2') You need $\Theta(|V|^3)$ space to store the adjacency matrix for graph $G = (V, E)$.

◯ True ◯ False

(b) (2') In a DFS traversal of a directed graph, if u is visited before v is visited, then there must exist a path from u to v.

◯ True ◯ False

(c) (2') Both Kruskal's and Prim's algorithms cannot correctly find the MST when there are negative edges in the graph.

◯ True ◯ False

(d) (2') Given a directed acyclic graph G. If u appears before v in a topological sort, there will not exist a path from v to u in G.

◯ True ◯ False

(e) (2') In A* graph search algorithm with a consistent heuristic function, if vertex u is marked visited before v, then $d(u) + h(u) \leq d(v) + h(v)$, where $d(u)$ is the distance from the start vertex to u.

◯ True ◯ False

(f) (2') Dijkstra's algorithm can work correctly on any graphs with negative edges.

◯ True ◯ False

(g) (2') The run time complexity of Floyd-Warshall algorithm can be reduced to $O(|V|^2)$ if we only need to compute the shortest path between a given pair of nodes instead of the shortest paths between all pairs of nodes.

◯ True ◯ False

(h) (2') We want to count the number of connected components in an undirected graph using disjoint set. We union the two vertices of each edge. Then the number of vertices such that $\mathtt{find}(v) = v$ is the answer.

◯ True ◯ False

(i) (2') If problem A polynomial-time reduces to problem B, then $B \in \mathsf{P}$ **only if** $A \in \mathsf{P}$.

◯ True ◯ False

(j) (2') If you can prove that there exists an NP problem that is not an $\mathsf{NP\text{-}Complete}$ problem, then you've proved $\mathsf{P} \neq \mathsf{NP}$.

◯ True ◯ False

2. (12 points) Single Choice

Each question has **exactly one** correct answer. Write your answers in the **answer sheet**.

(a) (3') Consider a simplest disjoint set of size 6 - with **no** path compression and **no** union by rank. The disjoint set has 2 operations:

- `find(int i)`: Find the root element of the tree that contains `i`.
- `union(int i, int j)`: Find the root elements of `i` and `j`, update `i`'s root to be `j`'s root.

Now given a sequence of operations, `union(1, 2)`, `union(3, 4)`, `union(5, 4)`, `find(3)`, `union(5, 1)`, `union(6, 3)`. What is the height of the disjoint-set tree that contains `2`?

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

(b) (3') If we use Dijkstra's Algorithm with binary heap optimization to find the shortest path from `s` to `t` on a positive-weighted graph, what is the **earliest** time that we ensure `dist[t]` is the real length of the shortest path?

A. When `t` is pushed into the heap for the first time.
B. When the heap is empty.
C. When we modify `dist[t]` for the first time.
D. When `t` is popped from the heap for the first time.

(c) (3') Which of the following statements about topological sort is correct?

1. Topological sort can be applied to any connected graph.
2. Topological sort is used to find (one of the) shortest paths in a weighted graph.
3. Topological sort is a linear ordering of vertices in a directed acyclic graph, where for every directed edge uv from vertex u to vertex v, u comes before v in the ordering.
4. Topological sort can result in multiple valid orderings for the same graph.

A. Only 1 and 2
B. Only 3 and 4
C. Only 1, 2 and 4
D. All of the above

(d) (3') The statement "2-Color is in NP-Complete." is ______?

A. True regardless of whether P equals to NP or not.
B. False regardless of whether P equals to NP or not.
C. True if and only if $\mathsf{P} = \mathsf{NP}$.
D. True if and only if $\mathsf{P} \neq \mathsf{NP}$.

3. (20 points) Multiple Choices

Each question has **one or more** correct answer(s). Select all the correct answer(s). For each question, you will get 0 points if you select one or more wrong answers, but you will get 2.5 points if you select a non-empty subset of the correct answers. Write your answers in the **answer sheet**.

(a) (5') Let G be a weighted undirected graph with positive weights where edge e has weight $w_e \in \mathbb{R}^+$ for all $e \in \mathsf{E}$. And G' is a copy of G except that edge e has weight $f(w_e)$. Which of the following statements is/are true?

A. If $f(w_e) = w_e + \frac{4}{w_e}$, then any MST in G is also an MST in G'.

B. If $f(w_e) = \exp(w_e + w_e^3)$, then any MST in G is also an MST in G'.

C. If $f(w_e) = |w_e \log(w_e)|$, then any MST in G is also an MST in G'.

D. If $f(w_e) = 1 - \frac{1}{1+w_e^8}$, then any MST in G is also an MST in G'.

(b) (5') Which of the following statements about shortest path algorithms is/are true?

A. If we modify the outer loop of Bellman-Ford algorithm to execute $|V|$ iterations instead of $|V| - 1$ iterations, the algorithm can still find the shortest path on a directed graph with negative-weight edges but no negative cycles.

B. If we modify the outer loop of Floyd-Warshall algorithm to execute $|V| - 1$ iterations instead of $|V|$ iterations, the algorithm can still find all pairs of shortest paths on a directed graph with negative-weight edges but no negative cycles.

C. We can modify the Bellman-Ford algorithm to find the longest path in a directed graph with positive-weight edges but no positive cycles.

D. We can modify Floyd-Warshall algorithm to detect whether there exists a negative cycle or not in a directed graph.

(c) (5') Given an undirected graph $\mathsf{G} = (V, \mathsf{E})$ whose adjacency matrix is A ($\forall\, u, v \in V$, $A_{uv} = 1$ if $\{u, v\} \in \mathsf{E}$ and 0 otherwise). Which of the following statements is/are true?

A. If G is connected, then $|\mathsf{E}| \geq |V| - 1$.

B. If G is acyclic, then $|\mathsf{E}| \leq |V| - 1$.

C. $\sum_{u,v \in V} A_{uv} = |\mathsf{E}|$.

D. $\forall\, u \in V$, $(A^2)_{uu} = \mathsf{degree}(u)$.

(d) (5') Which of the following statements is/are true?

A. 3-SAT can be reduced to 3-Color in polynomial time, and vice versa.

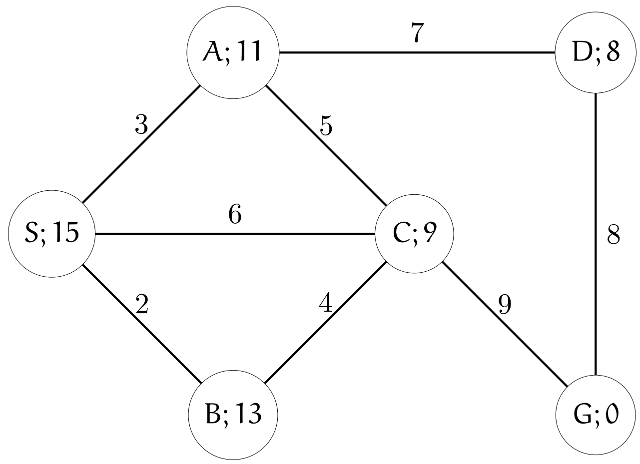
B. A problem A is NP-Complete if every problem in NP can be reduced to A in polynomial time.

C. If you prove 3-SAT can be reduced to 2-SAT in polynomial time, then you've proved $\mathsf{P} = \mathsf{NP}$.

D. If you prove that any two problems in NP can be reduced to each other in polynomial time, then you've proved $\mathsf{P} = \mathsf{NP}$.

4. (12 points) Graph theory benchmark

Consider the weighted undirected graph shown in the figure where G is the goal vertex. When choosing which vertex to visit next, always visit in **alphabetical** order if there is a tie. For A* Search algorithm, the heuristic function is written inside the node. For example, $h(S) = 15$.



(a) **Fill in the blanks**

i. (1') The degree of vertex C is ____________.

ii. (1') Is (S, C, A, D, G, C, B) a simple path? Yes/No: ____________.

iii. (2') Is the heuristic function admissible? ____________. Is it consistent? ____________.

(b) **MST**

i. (2') If we run Kruskal's algorithm on this graph, what are the edges in the MST? **Write their weights** in the order we added them.

ii. (2') If we run Prim's algorithm on this graph starting from B, what is the order of vertices we mark visited?

(c) **Shortest Path**

Note: In this question, we only care about the shortest path from S to G, so we will terminate the algorithm once we find the shortest path.

i. (2') Suppose we run Dijkstra's algorithm on this graph starting from S. Write down the order of vertices we marked visited.

ii. (2') If we run A* graph search algorithm on this graph from S, write down the order of vertices we marked visited.

5. (8 points) Another house coloring problem

You need to paint n houses in a row with m colors.

You will have a cost $c_{i,j}$ if you paint the i-th house with the j-th color.

You will have an additional cost b_j for each two consecutive houses painted with the same j-th color.

For example, if you paint $n = 6$ houses with colors 122233, your total cost is

$$c_{1,1} + c_{2,2} + c_{3,2} + c_{4,2} + c_{5,3} + c_{6,3} + 2b_2 + b_3$$

Please design a **dynamic programming** algorithm that returns the minimum total cost.

(a) (2') How will you define the subproblems?

(b) (1') What is the answer to this question in terms of your subproblems?

(c) (4') Give your Bellman equation to solve the subproblems.

(d) (1') What is the runtime complexity of your algorithm? (answer in $\Theta(\cdot)$)
Prefix/suffix optimization is not required.

6. (8 points) Interval Covering Problem

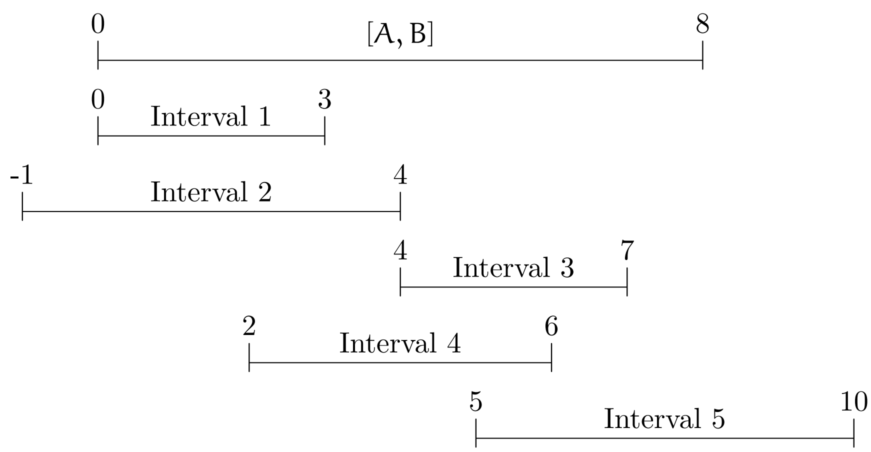
There are n intervals. The i-th interval is $[l_i, r_i]$. Your target is to fully cover the range $[A, B]$ with as few intervals as you can.

It is promised that $\forall x \in [A, B], \exists i \in [1, n], l_i \leqslant x \leqslant r_i$, which means that a solution that covers $[A, B]$ always exists.

Example: You need at least 3 intervals to fully cover $[A, B] = [0, 8]$ by these $n = 5$ intervals:

$[0, 3], [-1, 4], [4, 7], [2, 6], [5, 10]$, and there are multiple solutions that uses 3 intervals:

- $[0, 3], [2, 6], [5, 10]$
- $[-1, 4], [2, 6], [5, 10]$
- $[-1, 4], [4, 7], [5, 10]$



Please design a **greedy algorithm** to find one solution with the fewest intervals.

(a) (3') Describe your algorithm in **natural language** or **pseudocode**.

(b) (1') The time complexity of this algorithm is ______________ (in terms of $O(f(n))$).

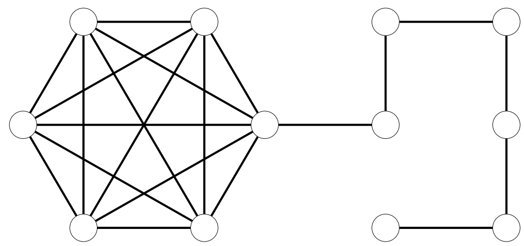
(c) (4') Prove the correctness of your greedy algorithm by **exchanging arguments**.

7. (8 points) Kite is in NP-Complete

In this question, we will prove that Kite is in NP-Complete.

Recall that a **clique** of size k is a graph on k vertices such that any two of them are adjacent (i.e. all the k vertices are connected to each other).

Based on this, we define that a **kite** of size k is a graph on $2k$ vertices such that k of the vertices form a clique and the remaining k vertices are connected in a "tail" that consists of a path joined to one of the vertices of the clique. Below is a kite of size 6.



Then consider the following problem:

Kite: Given a undirected graph $G = (V, E)$ and a positive integer $k \geq 3$, determine G contains a kite of size at least k. (To avoid corner cases here, we assume $k \geq 3$.)

The yes-instances of Kite is:

$$\mathsf{Kite} = \left\{ \langle G, k \rangle \;\middle|\; \begin{array}{l} k \geq 3 \text{ and } G = (V, E) \text{ is an undirected graph} \\ \text{that contains a kite of size at least } k. \end{array} \right\}$$

(a) (2') Prove that Kite is in NP. (Show your certificate and certifier.)

(b) (0') We choose Clique to reduce from. Recall that the yes-instance of Clique is:

$$\mathsf{Clique} = \left\{ \langle G', k' \rangle \;\middle|\; \begin{array}{l} k' \geq 3 \text{ and } G' = (V', E') \text{ is an undirected graph that contains} \\ k' \text{ vertices such that they are connected to each other.} \end{array} \right\}$$

(Note that here we also assume $k' \geq 3$ to avoid corner cases.)

(c) (3') Construct your **correct** polynomial-time many-one reduction f that maps instances of Clique to instances of Kite.

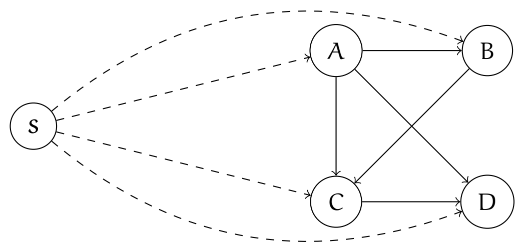
(d) Prove the correctness of your reduction by showing:

i. (1') x is a yes-instance of Clique $\Rightarrow$ $f(x)$ is a yes-instance of Kite.

ii. (2') x is a yes-instance of Clique $\Leftarrow$ $f(x)$ is a yes-instance of Kite.

8. (12 points) All-Pairs Shortest Paths on Sparse Graphs

Let $G = (V, E)$ be a simple directed graph that does not contain a negative cycle. Let $w(u, v) \in \mathbb{R}$ be the weight of the edge $(u, v) \in E$. We make a new graph $G' = (V', E')$ where $V' = V \cup \{s\}$ for a new vertex $s \notin V$, and $E' = E \cup \{(s, v) \mid v \in V\}$. The construction of the new graph is demostrated as follows.



Let $w'(u, v)$ be the weight of the edge $(u, v) \in E'$ on the new graph, defined as

$$w'(u,v) = \begin{cases} w(u,v), & \text{if } (u,v) \in E, \text{ i.e. } u \neq s, \\ 0, & \text{if } u = s. \end{cases}$$

Let $h(v)$ be the length of the shortest path on G' from s to v, where $v \in V$.

(a) (2') What is the most efficient algorithm to compute $h(v)$ for all $v \in V$? Choose only one choice. ____________.

A. Dijkstra B. Bellman-Ford C. Floyd-Warshall D. Prim

(b) (2') Prove that $h(u) + w(u, v) \geqslant h(v)$ for every $(u, v) \in E$.

(c) (4') For each edge $(u, v) \in E$ on the original graph G, we define a new weight function as

$$\hat{w}(u,v) = w(u,v) + h(u) - h(v)$$

so that $\hat{w}(u, v) \geqslant 0$. Let $p = \langle v_0, v_1, \cdots, v_k\rangle$ be any simple path from v_0 to v_k, where $v_0, \cdots, v_k \in V$. Prove that if p is a shortest path from v_0 to v_k with weight function $\hat{w}$, then it is also a shortest path from v_0 to v_k with weight function w.

Hint: Let $w(p)$ and $\hat{w}(p)$ be the length of p with weight function w and $\hat{w}$ respectively. Consider $\hat{w}(p) = \sum_{i=1}^{k} \hat{w}(v_{i-1}, v_i)$.

(d) (4') Now we want to find the lengths of shortest paths between all pairs of vertices in G. If $|E| = \Theta(|V|)$ (which means that the graph is *sparse*), can you come up with an algorithm that solves this problem with time complexity asymptotically better than the Floyd-Warshall algorithm? Briefly describe your algorithm in **natural language** or **pseudocode**, and give its time complexity.

Hint: With weight function $\hat{w}$, the graph does not contain negative weight edges. How can you make use of this property?

This page is for your sketch.

This page is for your sketch.

Name: ID:

1 True or False

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)

2 Single Choice

(a)	(b)	(c)	(d)

3 Multiple Choice

(a)	(b)	(c)	(d)

4 Graph theory benchmark

(a)

i. The degree of vertex C is ______________.

ii. Is $(\mathsf{S}, \mathsf{C}, \mathsf{A}, \mathsf{D}, \mathsf{G}, \mathsf{C}, \mathsf{B})$ a simple path? Yes/No: ______________.

iii. Is the heuristic function admissible? ______________. Is it consistent? ______________.

(b)

i.

ii.

(c)

i.

ii.

5 Another house coloring problem

(a)

(b)

(c)

(d)

6 Interval Covering Problem

(a)

(b) The time complexity of this algorithm is ____________________ (in terms of $O(f(n))$).

(c)

You can continue to write your answer of (c) in the next page.

7 Kite is in NP-Complete

(a)

(b) You don't need to answer this question.

(c)

(d)

i.

ii.

8 All-Pairs Shortest Paths on Sparse Graphs

(a) ______________.

A. Dijkstra B. Bellman-Ford C. Floyd-Warshall D. Prim

(b)

(c)

(d)