

CMSC 451 Final Exam

This exam is closed-book and closed-notes. You may use two sheets of notes (front and back). You may use any algorithms or results given in class. The total point value is 100 points. Good luck!

Problem 1. (30 points) Explanations are not required, but may be provided for partial credit.

- (a) (5 points) This question involves the maximum/minimum number of edges in a graph with n vertices. Assume no multi- or self-loop edges.
- (1) **Maximum** number of edges in an **undirected** graph
 - (2) **Minimum** number of edges in a **connected** undirected graph
 - (3) **Maximum** number of edges in a **DAG**

For each selecting among the following options:

- (i) $n - 1$
 - (ii) n^2
 - (iii) $n(n - 1)$
 - (iv) $n(n - 1)/2$
 - (v) None of these
- (b) (5 points) Let G be a directed graph and suppose that we perform a DFS on G . For each vertex u , let $d[u]$ and $f[u]$ denote its discovery and finish times, respectively. Suppose that (u, v) is a *back edge*. Which of the following is necessarily true (select all that apply).
- (i) $d[u] < d[v]$
 - (ii) $d[u] > d[v]$
 - (iii) $f[u] < f[v]$
 - (iv) $f[u] > f[v]$
 - (v) G is *not* a DAG
- (c) (8 points) You are given two strings X and Y where $|X| = m$ and $|Y| = n$. Suppose that the length of their longest common subsequence (LCS) is k and the length of the shortest common supersequence is j . Which of the following necessarily hold. (Select all that apply.)
- (i) $k \leq \min(m, n)$
 - (ii) $j \geq \max(m, n)$
 - (iii) $j + k = m + n$
 - (iv) If $k = \min(m, n)$, then one string is a subsequence of the other
- (d) (2 points) Suppose the Floyd-Warshall algorithm is run on a graph that contains a *negative-cost cycle*. Which of the following best describes what will occur?
- (i) It will terminate with the correct answer

- (ii) It will terminate, but the answer may not be correct (or may be meaningless)
 - (iii) It will go into an infinite loop
 - (iv) It will abort
 - (v) None of the above
- (e) (2 points) True or False: The *shortest duration first* (SDF) strategy for interval scheduling is optimal. (SDF means to schedule intervals in increasing order of duration $f_i - s_i$.)
- (f) (8 points) For each of the following claims, state whether it is (T) True, (F) False, or (U) Unknown to science. Let A and B denote two languages.
- (i) $P \subset NP$ (P is a *proper* subset of NP)
 - (ii) $NP \subset P$ (NP is a *proper* subset of P)
 - (iii) If $A \in NP$ and $B \in NP\text{-hard}$ then $A \leq_P B$
 - (iv) If $A \leq_P B$ and $B \leq_P C$ then $A \leq_P C$

Problem 2. (20 points) A rancher needs to transport his H horses to town, but he has no truck. A set of n contractors have offered to help transport the horses. For $1 \leq i \leq n$, the i th contractor can carry h_i horses in his truck, and will charge the rancher c_i dollars. Given H , h_i 's and c_i 's, the rancher wants to compute the cheapest set of trucks to transport all H horses. (You may assume that $\sum_i h_i \geq H$, so there is a feasible solution.)

- (a) (10 points) Suppose that the rancher only pays for the *fraction* of the truck that is used. For example, if the i th truck is half filled with horses, the rancher only pays $c_i/2$ (see Fig. 1).

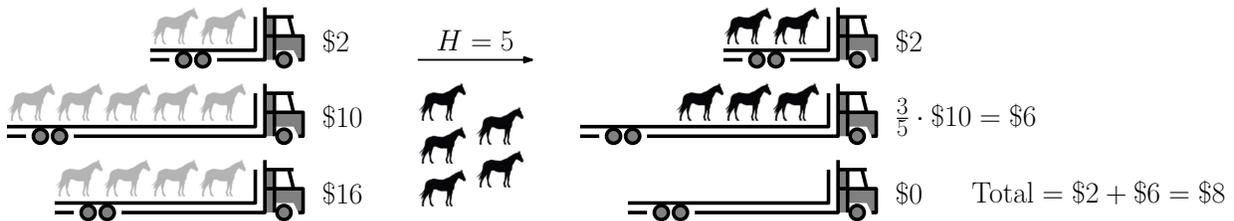


Figure 1: Filling trucks.

Present an efficient *greedy* algorithm that, given H , and the values h_i and c_i , for $1 \leq i \leq n$, determines how many horses to place in each truck in order to minimize the total cost. (Briefly explain. You don't need to prove correctness.)

- (b) (10 points) Repeat (a) but with a different cost model. Now, the rancher must pay the *full price* of each truck used, even if there is only one horse on the truck (see Fig. 2). Present an efficient algorithm for this problem. (**Hint:** Use DP. It suffices to give the recursive formulation to determine the optimal cost. Remember to include the basis case and how to obtain the final answer. Aim for a running time of $O(nH)$.)

Problem 3. (15 points) You are given a sequence of n buckets, indexed 0 through $n - 1$, holding a total of m balls. You are given two sequences of nonnegative integers $A = \langle a_0, \dots, a_{n-1} \rangle$ and $B = \langle b_0, \dots, b_{n-1} \rangle$, where $\sum_i a_i = \sum_i b_i = m$. Initially, the i th bucket holds a_i balls, for

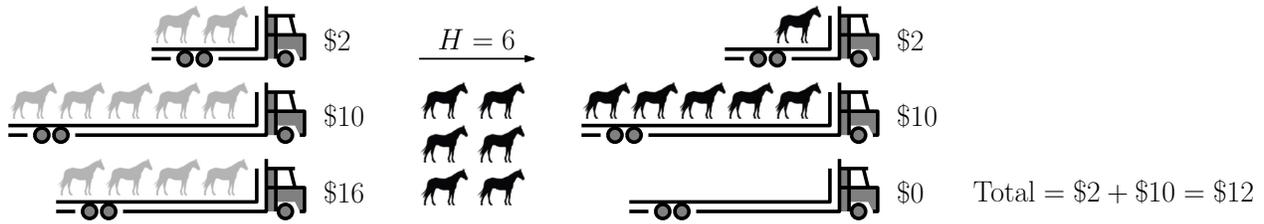


Figure 2: Filling trucks.

$0 \leq i \leq n - 1$. Your objective is to determine whether it is possible to redistribute the balls so that after, the i th bucket holds b_i balls, for $0 \leq i \leq n - 1$. The redistribution must satisfy the following requirements:

- Each ball can remain in the same bucket, or it can move at most two buckets to the right. (That is, a ball in bucket i can stay in bucket i or move to any of the buckets $\{i + 1, i + 2\}$. Indices are taken mod n , so they wrap around to the start.)
- At most 3 balls can move from any one bucket to any *different* bucket.
- For each bucket, at least 2 of its original balls must remain in this bucket. (You may assume $a_i \geq 2$, for all i .)

An example of an input and a redistribution is shown in Fig. 3.

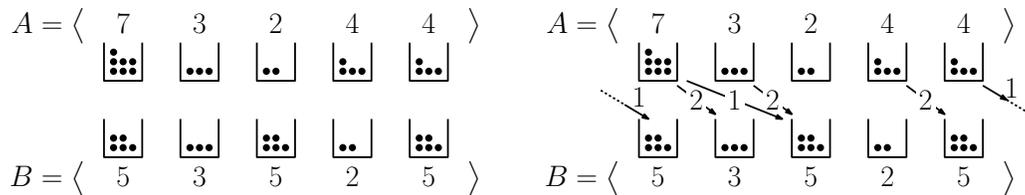


Figure 3: Ball redistribution.

- Explain how to model this problem as a circulation problem (with lower and upper edge capacities). Briefly explain your construction. (We do *not* want a full correctness proof.)
- Illustrate your construction on the following input: $A = \langle 5, 3, 2, 4 \rangle$ and $B = \langle 3, 5, 4, 2 \rangle$. (Just present the construction of the circulation network, you do not need to show the final flow values.)

Problem 4. (20 points) For any integer $p \geq 1$, the graph $K_{p,p}$ is a complete bipartite graph with p vertices in half of the graph (see Fig. 4(a) for the graph $K_{3,3}$). The *complete bipartite subgraph problem* (CBS) is the following: Given a graph $G = (V, E)$ and an integer $p \geq 1$, does G have a copy of $K_{p,p}$ as a subgraph? That is, is there a subset $V' \subseteq V$ of size $2p$, such that the subgraph of G induced on these vertices is equal to $K_{p,p}$? (See Fig. 4(b).) We will show that CBS is NP-complete

- (10 points) Prove that CBS is in NP. (What is the certificate and how do you test that it satisfies the CBS condition?)

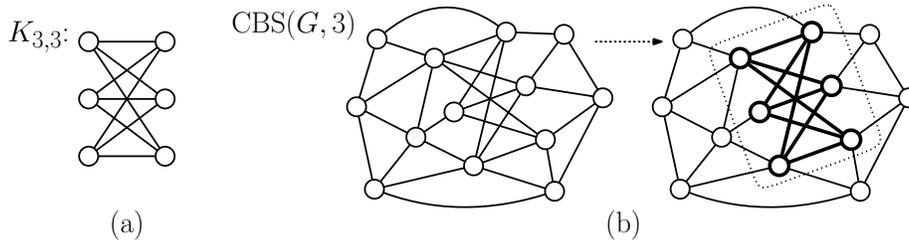


Figure 4: Complete bipartite subgraph (CBS).

- (b) (10 points) Prove that CBS is NP-hard by showing that some known NP-complete problem is reducible to it. (**Hint:** Reduce from independent set (IS), but you may use any NP-Complete problem covered in class.) Present your reduction and provide a short explanation (not a full correctness proof).

Problem 5. (15 points) Given an undirected graph $G = (V, E)$, consider the following simple greedy heuristic for computing an *independent set of maximum size* in G :

- $S \leftarrow \{\}$
- while G is not empty:
 - select the vertex u having the smallest degree (with ties broken arbitrarily)
 - add u to S
 - delete u , all its neighbors, and all incident edges from G
- return S

(An example is shown in Fig. 5.)

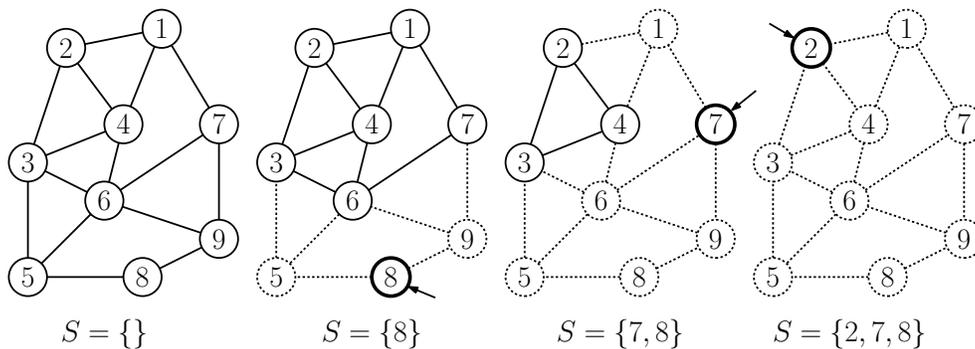


Figure 5: Greedy independent set heuristic.

- (a) (10 points) Prove that if every vertex of G has *degree at most 4*, then the above heuristic outputs an independent set S of size at least $n/5$, where $n = |V|$.
- (b) (5 points) Briefly explain why (a) implies that, for graphs of degree at most 4, this greedy heuristic has an *approximation ratio* of 5.