

CSCI-570 Analysis of Algorithms

Practice Exam - 2

2020.4.24

True/False Problems

1. For every graph G and every maximum flow on G , there always exists an edge such that increasing the capacity on that edge will increase the maximum flow that's possible in the graph.
2. The problem of deciding whether a given flow f of a given flow network G is maximum flow can be solved in linear time.
3. In any maximum flow there are no cycles that carry positive flow. (A cycle $\langle e_1, \dots, e_k \rangle$ carries positive flow iff $f(e_1) > 0, \dots, f(e_k) > 0$.)
4. If X is any search problem, then X can be reduced to INDEPENDENT SET.
5. An linear program cannot have an unbounded objective function value.
6. The problem of finding the length of the shortest path between two nodes in a directed graph (which we would normally solve using Dijkstra's algorithm) can be expressed as a linear program. (An ordinary, continuous linear program; we're not talking about integer linear programs.) Assume that there is only one shortest path
7. Given a flow network where all the edge capacities are even integers, the Ford-Fulkerson algorithm will require at most $C/2$ iterations, where C is the total capacity leaving the source s .
8. Suppose f is a flow of some value X from s to t in a flow network G and there is an s - t cut of capacity X . Then there are no s to t paths in the residual graph G_f .
9. A graph flow is a max flow if and only if there exists no cut with the same capacity as the flow's value.
10. Adding a constraint to a linear programming problem increases the size of the feasible region.

Multiple Choice

1. The problem 3-SAT and 2-SAT are
 - (A) both in P
 - (B) both NP complete
 - (C) NP-complete and in P respectively
 - (D) undecidable and NP-complete respectively

2. Which of the following statements are TRUE? (1) The problem of determining whether there exists a cycle in an undirected graph is in P. (2) The problem of determining whether there exists a cycle in an undirected graph is in NP. (3) If a problem A is NP-Complete, there exists a non-deterministic polynomial time algorithm to solve A. (A) 1, 2 and 3
(B) 1 and 3
(C) 2 and 3
(D) 1 and 2

3. Which of the following is true about NP-Complete and NP-Hard problems.
 - (A) If we want to prove that a problem X is NP-Hard, we take a known NP-Hard problem Y and reduce Y to X
 - (B) The first problem that was proved as NP-complete was the circuit satisfiability problem.
 - (C) NP-complete is a subset of NP Hard
 - (D) All of the above

Answer the following questions based on:

A company manufactures two products X and Y. Each product has to be processed in three departments: welding, assembly and painting. Each unit of X spends 2 hours in the welding department, 3 hours in assembly and 1 hour in painting. The corresponding times for a unit of Y are 3, 2 and 1 hours respectively. The employee hours available in a month are 1,500 for the welding department, 1,500 in assembly and 550 in painting. The contribution to profits are £100 for product X and £120 for product Y.

4. What is the objective function (Z) to be maximised in this linear programming problem (where Z is total profit in £s)?
 - (A) $Z = 1500X + 1500Y$
 - (B) $Z = 120X + 100Y$
 - (C) $Z = 2X + 3Y$
 - (D) $Z = 100X + 120Y$

5. Total profits are maximised when the objective function (as a straight line on a graph) is:
 - (A) Furthest from the origin irrespective of the 'feasible region'
 - (B) Nearest to the origin and tangent to the 'feasible region'
 - (C) Furthest from the origin and tangent to the 'feasible region'
 - (D) Nearest to the origin irrespective of the 'feasible region'

6. What is the equation of the labour constraint line for the welding department in this linear program?
- (A) $3X + 2Y = 1,500$ hours
 - (B) $3X + 2Y = 550$ hours
 - (C) $2X + 3Y = 1,500$ hours
 - (D) $2X + 3Y = 550$ hours
7. What is the equation of the labour constraint line for the assembly department in this linear program?
- (A) $1X + 1Y = 1,500$ hours
 - (B) $2X + 2Y = 1,500$ hours
 - (C) $1X + 1Y = 550$ hours
 - (D) $3X + 2Y = 1,500$ hours
8. What is the solution to this linear programming problem in terms of the respective quantities of X and Y to be produced if profits are to be maximised?
- (A) $X = 150, Y = 400$
 - (B) $X = 0, Y = 500$
 - (C) $X = 400, Y = 150$
 - (D) $X = 550, Y = 0$

Problem LP

Given the following linear program

$$\max(3x_1 + 8x_2)$$

subject to

$$x_1 + 4x_2 \leq 20$$

$$x_1 + x_2 \geq 7$$

$$x_1 \geq -1$$

$$x_2 \leq 5$$

1. Rewrite it in the standard form. You don't need to write in matrix form.
2. Write the dual.

Problem NPC

Suppose that after your graduation, you move to a new city. The city is defined by a directed graph $G = (V, E)$ and for each edge $e \in E$, there is a non-negative cost c_e associated to it. Your living place is represented as a node $s \in V$. In addition, there are a set of landmarks $X \subseteq V$, which you want to visit. Therefore, you want to choose a subgraph that spans all vertices in X (and possibly some other in V) and contains a path from s to each $x \in X$ with the minimum total edge cost.

1. Phrase the above optimization problem as a decision problem and show that it belongs to NP.
2. Show a polynomial time construction using a reduction from 3-SAT.
3. Write down the claim that the 3-SAT problem is polynomially reducible to the original problem.
4. Prove the claim in the direction from 3-SAT to the reduced problem.
5. Prove the claim in the direction from the reduced problem to 3-SAT.

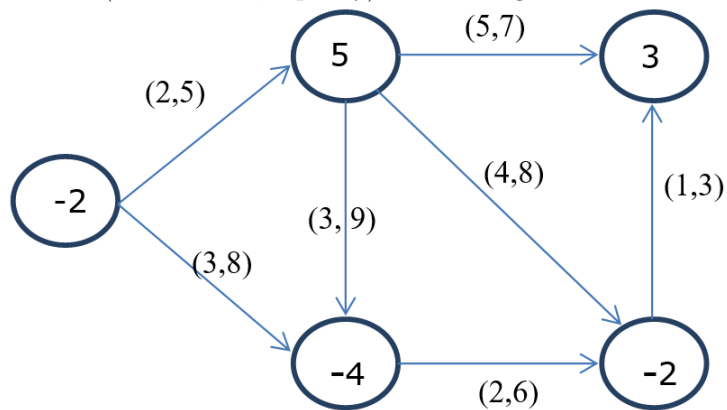
Problem DP

Given a sequence of n coins of values v_1, \dots, v_n , where n is even. We play a game against an opponent by alternating turns. In each turn, a player selects either the first or last coin from the sequence, removes it, and receives the value of that coin. Use dynamic programming to determine the maximum possible amount of money we can win if we move first.

1. Define (in plain English) subproblems to be solved.
2. Write the recurrence relation for subproblems. Also write the base cases.
3. Use plain English explain how you use that recursive relation to obtain a solution. You don't need to prove the correctness.
4. State the runtime of the algorithm. Explain your answer.

Problem Network Flow 1

In the network below, the demand values are shown on vertices. Lower bounds on flow and edge capacities are shown as (lower bound, capacity) for each edge.



Answer the following questions:

1. Describe how to reduce the circulation problem into a maximum flow problem.
2. Compute the maximum flow of the reduced problem.
3. Determine if there is a feasible circulation in the original graph.

Problem Network Flow 2

Suppose we have a binary matrix $A \in \{0,1\}$ of size $m \times n$. Our goal is to select rows and columns that contain 1's. That is, each 1 should be covered by either a column or a row. Each row have a cost r_i and each column have a cost c_j . Design an algorithm to find the lowest total cost:

$$\min \prod_{i \in [m]} r_i \prod_{j \in [n]} c_j$$