# CS570 Spring 2022: Analysis of Algorithms Exam III

| | Points | | Points |
|---|---|---|---|
| Problem 1 | 20 | Problem 5 | 9 |
| Problem 2 | 9 | Problem 6 | 15 |
| Problem 3 | 18 | Problem 7 | 17 |
| Problem 4 | 12 | | |
| | **Total** | **100** | |

Instructions:
1. This is a 2-hr exam. Open book and notes. No electronic devices or internet access.
2. If a description to an algorithm or a proof is required, please limit your description or proof to within 150 words, preferably not exceeding the space allotted for that question.
3. No space other than the pages in the exam booklet will be scanned for grading.
4. If you require an additional page for a question, you can use the extra page provided within this booklet. However please indicate clearly that you are continuing the solution on the additional page.
5. Do not detach any sheets from the booklet. Detached sheets will not be scanned.
6. If using a pencil to write the answers, make sure you apply enough pressure, so your answers are readable in the scanned copy of your exam.
7. Do not write your answers in cursive scripts.
8. This exam is printed double sided. Check and use the back of each page.

1) 20 pts
Mark the following statements as **TRUE** or **FALSE** by circling the correct answer. No need to provide any justification.

**[ TRUE/FALSE ]**
If P = NP, then all problems in P are NP-complete.

**[ TRUE/FALSE ]**
Assuming that we have found a correct optimal solution to an LP problem, if someone claims to have found a different optimal solution, we know that that solution must be incorrect.

**[ TRUE/FALSE ]**
Let $X$ and $Y$ be decision problems for which there exist efficient certifiers, and assume P ≠ NP. If $X \leq_P Y$ and $Y \leq_P X$, then both $X$ and $Y$ are NP-complete.

**[ TRUE/FALSE ]**
Minimum Spanning Tree (MST) $\leq_P$ Hamiltonian Path.

**[ TRUE/FALSE ]**
The optimization version of the Linear Programming is not in NP.

**[ TRUE/FALSE ]**
If P=NP, then some NP-hard problems can be solved in polynomial time.

**[ TRUE/FALSE ]**
Even if P≠NP, there is a polynomial-time algorithm for the 0/1 Knapsack problem when all items have the same weight-to-value ratio

**[ TRUE/FALSE ]**
If P = NP, then every decision problem whose solution can be verified in polynomial time can itself be solved in polynomial time.

**[ TRUE/FALSE ]**
Integer Max Flow (where flows and capacities are integer-valued) is polynomial-time reducible to Linear Programming.

**[ TRUE/FALSE ]**
Maximize $\Sigma\, x_i(1 - x_i)$ $\quad (1 \leq i \leq n)$
subject to $x_i + x_j < 1$ and $x_i \in \{0, 1\}$ $\quad (1 \leq i \leq n,\ 1 \leq j \leq m)$
where $n$ and $m$ are constants, is an integer linear program.

2) 9 pts

**i)** Which of the following problems have been proved to be not solvable in polynomial time? Circle all correct answers

A. Traveling Salesman
B. Integer Linear Programming
C. Subset sum
D. All of the above
E. None of the above

**ii)** Let *X, Y, Z,* and *W* be problems. Suppose *Z* is polynomial-time reducible to *X*, and *X* is NP-complete. Which of the following statements are true? Circle all correct answers.

A. If Vertex Cover is polynomial-time reducible to problem *Z*, then *Z* is NP-Complete.
B. If there exists an efficient certifier for problem *Z*, then *Z* is NP-Complete.
C. *X* is polynomial-time reducible to Independent Set.
D. If *X* is polynomial-time reducible to *Y*, then *Z* is polynomial-time reducible to *Y*.
E. If *Z* is polynomial-time reducible to *W*, then *X* is polynomial-time reducible to *W*.

**iii)** If a problem *X* is in NP, then it must be NP-complete if… (Circle all correct answers)

a. It is polynomial-time reducible to 3-SAT.
b. 3-SAT can be reduced to it in polynomial time.
c. It can be reduced to every other problem in NP in polynomial time.
d. Some problem in NP can be reduced to it in polynomial time.

3) 18 pts

A paper company is building warehouses to supply its $m \geq 1$ customers. The company is considering whether to build a warehouse at each of $n \geq 1$ potential construction sites. The cost of building a warehouse at site $i$ is denoted by $b_i \geq 0$. After the warehouses are built, the company assigns each customer to a warehouse that will supply it. Each warehouse that is built may supply paper to any number of customers, but each customer is supplied by exactly one warehouse. The cost of supplying customer $j$ from a warehouse built at site $i$ is denoted by $s_{ij} \geq 0$.

Consider the Warehouse Construction Optimization problem:
*Given construction costs $b_i$ $(1 \leq i \leq n)$ and warehouse-customer supply costs $s_{ij}$ $(1 \leq i \leq n,\ 1 \leq j \leq m)$, make a selection of a subset of construction sites and an assignment of customers to warehouses such that the total cost of building a warehouse at each selected site and supplying each customer from their assigned warehouse is minimized.*

a) Reduce the above Warehouse Construction problem to Integer Linear Programming. (16 pts total)

   i) Describe what your ILP variables represent. (5 pts)

   ii) Show your objective function. (4 pts)

   iii) Show your constraints (7 pts)

b) **[ TRUE/FALSE ]** (2 pts)
The above reduction shows that the Warehouse Construction optimization problem is NP-hard.

4) 12 pts
Suppose that we have developed an approximation algorithm that takes as input a graph $G=(V,E)$ and a real number $\rho > 1$ and returns a vertex cover whose size is less than $\rho$ times the size of a minimum vertex cover.

Prove that the running time of this approximation algorithm is not polynomial in the size of the input unless P=NP.

5) 9 pts

For each of the problems defined below, determine which class the problem belongs to from the options below. Assume P ≠ NP.

1. The problem is in P.
2. The problem is in NP but not in P.
3. The problem is neither in P nor NP.

a) Given a list of positive integers *A[1..n]* and a positive integer *k*, does there exist an increasing subsequence of *A* of length at least *k*?

b) Given an undirected graph with weights on its vertices, return an independent set with maximum total weight.

c) Given a computer program and an input to the program, will the program terminate for the given input?

6) 15 pts

Several artificial lights are installed in a greenhouse to provide light for several plants. Each light may provide light to multiple plants, and each plant receives sufficient light so long as it is exposed to at least one light. The greenhouse operators are planning to uninstall some lights to reduce their costs.

The greenhouse operators have designed a greedy algorithm for minimizing the number of lights in the greenhouse such that each plant is still exposed to at least one light:

> *Begin by turning off all the lights. Then, in each iteration, turn on the light that exposes the most plants to light that are not exposed yet. Repeat until all plants are exposed to at least one light. Then, uninstall the lights that are still turned off.*

a) Give a counterexample to prove that this algorithm does not always give the optimal solution. (5 pts)

b) Prove that if we used the above algorithm as an approximation algorithm, this approximation algorithm has no constant approximation ratio > 1. In other words, if the minimum number of lights needed is $C^*$, prove that there is no constant $\rho$ ($\rho > 1$) such that we can guarantee the algorithm returns a solution with total number of lights $\leq \rho C^*$. (10 pts)

7) 17 pts

Consider the following problem:

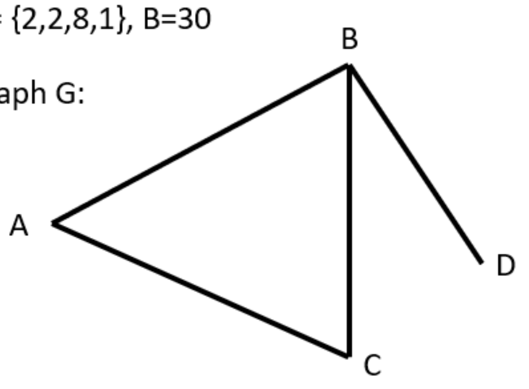
> *Given an undirected graph* G(V,E) *with* n *vertices, a list* L[1..n] *of* n *non-negative integers, and a non-negative integer* B*, does there exist a function* f : L → V *that assigns each number in* L *to a vertex in* G *such that*

$$\sum_{(u,v) \in E} f(u) \cdot f(v) \leq B?$$

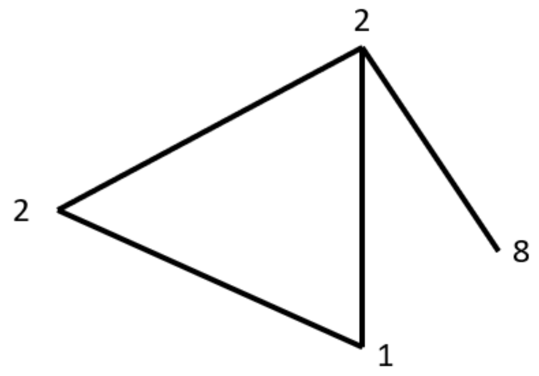
For example, the following is a YES instance of this problem (i.e. an instance for which such a function f exists):

L = {2,2,8,1}, B=30

Graph G:



Here is one possible assignment function:
$L_1 \rightarrow A, L_2 \rightarrow B, L_3 \rightarrow D, L_4 \rightarrow C$



$\sum_{(u,v) \in E} f(u) \cdot f(v) = (2*2) + (2*1) + (2*1) + (2*8) = 24 \leq 30$

Continued of next page

a) Prove that the problem is in NP. (5 pts)

b) Prove that the problem is NP-Complete. (Hint: use Vertex Cover) (12 pts)

Additional Space

Additional Space

Additional Space