This page is intentionally left blank.
Please do not open this question set before you are allowed to do so.

4810-1183 Approximation and Online Algorithms with Application (Spring 2017) #2

Quiz Problem 1

In this problem, you will construct a problem from an optimization model, then prove that it is an NP-hard problem.

Consider the following situation.

> We want to monitor all communications in the social network as always. However, because monitoring all communications is too costly, we allow 3 communications to be unobserved. (So, if we have m communications in the social network, we want to observe at least $m - 3$ communications)

Question 1.1: State inputs of this problem by a mathematical formulation.

Question 1.2: State outputs of this problem by a mathematical formulation.

Question 1.3: State constraints of this problem by a mathematical formulation.

Question 1.4: State objective functions of this problem by a mathematical formulation.

Question 1.5: Write a program for solving vertex cover problem based on the fact that an efficient algorithm for solving the optimization model in Problems 1.1 – 1.4 is given in a library.

```
[AnswerOf1_2] YourOptimizationModel([AnswerOf1_1]);

Sets VertexCover(Set V, Set E){
     //write your code for knapsack here
}
```

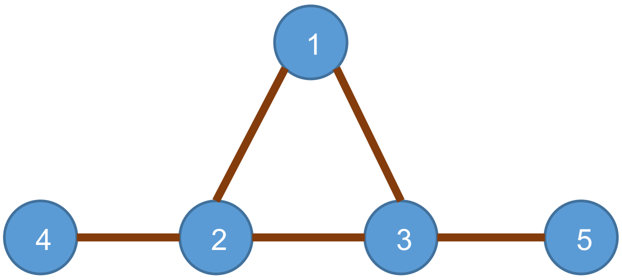
Question 1.6: Discuss why your optimization model is NP-hard based on your answer of Question 1.5.

4810-1183 Approximation and Online Algorithms with Application (Spring 2017) #2

Quiz Problem 2

We will continue working on your optimization model in Problem 1 in this problem. We will try using deterministic rounding for solving the model. Answering Question 1.1 – 1.4 is a prerequisite for this problem.

Question 2.1: Suppose that the number of unobserved communication is 1. What is the optimal solution and optimal value for your optimization model when the social network is as follows?



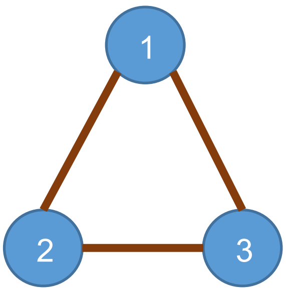
From next question, we refer to a linear program such that an output $\boldsymbol{x}$ must be a vector of integers as an integer program. In other words, given matrix $\boldsymbol{A}$, vector $\boldsymbol{b}$, and vector $\boldsymbol{c}$, we want to find an integer vector $\boldsymbol{x}$ such that $\boldsymbol{Ax} \geq \boldsymbol{b}$ and $\boldsymbol{c}^t\boldsymbol{x}$ is minimized.

Question 2.2: Construct an integer program to solve an example in Question 2.1. What is matrix $\boldsymbol{A}$, vector $\boldsymbol{b}$, and vector $\boldsymbol{c}$ for this case? What is the optimal solution obtain from the linear program?

Question 2.3: Generalize Question 2.2. Give an idea how to construct matrix $\boldsymbol{A}$, vector $\boldsymbol{b}$, and vector $\boldsymbol{c}$ from an arbitrary input of this problem. (You need not to really construct the matrix and vectors. Just give the idea how to construct them.)

From next question, we will consider your integer program in Question 2.3. However, we will not consider the condition that $\boldsymbol{x}$ has to be an integer vector.

Question 2.4: Consider the following social network. Suppose that the number of unobserved communication is 1.



Explain why assigning 1/3 to all nodes in the output also provides a feasible solution to your linear program.

Question 2.5: Explain why deterministic rounding in the class does not work for this problem.

Question 2.6: Devise a deterministic rounding algorithm for this problem.
Hint: Your algorithm need not to be 2-approximation algorithm and you can consider the number of unobserved communication during the rounding process.

4810-1183 Approximation and Online Algorithms with Application (Spring 2017) #2

Quiz Problem 3

During a class, we discuss a greedy algorithm that is a 2-approximation algorithm for knapsack problem. In this problem, we will show that the approximation ratio can be improved, when we assume that the weights of all strawberries are smaller than 10% of the maximum weight we can eat. (which is a practical assumption)

Question 3.1: In the greedy algorithm, we assume that $\frac{h_1}{w_1} \geq \frac{h_2}{w_2} \geq \cdots \geq \frac{h_n}{w_n}$. Then, we eat strawberry 1, 2, 3, … until we cannot eat more strawberry. Suppose that the last strawberry we eat is strawberry i. Discuss why $w_1 + \cdots + w_i \geq 0.9W$ when W is the maximum weight we can eat.

Question 3.2: In the greedy algorithm, if $h_{i+1} > h_1 + \cdots + h_i$ we will eat strawberry $i+1$ instead of strawberries $1, \ldots, i$. Discuss why $h_{i+1} > h_1 + \cdots + h_i$ is not true in all possible inputs.

Question 3.3: Discuss why $h_1 + \cdots + h_i \geq 9 \cdot h_{i+1}$.

Question 3.4: Discuss why $h_1 + \cdots + h_i + h_{i+1} \leq \frac{10}{9}(h_1 + \cdots + h_i)$.

Question 3.5: Discuss why $OPT \leq \frac{10}{9} \cdot SOL$

Question 3.6: What is the approximation ratio of our algorithm when all strawberries are smaller than 10% of the maximum weight? Can you guess the approximation ratio of our algorithm when all strawberries are smaller than x% of the maximum weight?