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

Final Problem 1

In this problem, we will devise an online algorithm for the following situation.

> There is a very strange pension system in Country A. Similar to many countries, when a person turns 60, he/she can choose to receive a lump sum or a yearly pension. However, there is one different rule. A person who is receiving the yearly pension can decide to receive the lump sum, but the lump sum amount will be deduced. The deduction amount is 2 times of the amount receiving as a yearly pension.
>
> For example, assume that the lump sum amount is 80 Million Yen, when a person receive it at 60, and the yearly pension is 8 Million Yen. If the person choose to receive the yearly pension at 60 but decide to receive the lump sum at 63, the lump sum amount that he/she can receive at 63 is
> 80 Million – 2 x (8 Million x 3 Years) = 24 Million.

From next question, assume that the lump sum amount is 80 Million Yen, when a person receive it at 60, and the yearly pension is 8 Million Yen.

<u>Question 1.1</u>: If we know that we live until 99, what is the best strategy for this pension system? How much can we have from the best strategy?

<u>Question 1.2</u>: If we choose the yearly pension at 60 but turn to choose the lump sum at 63, how much can we receive if we live until 99?

<u>Question 1.3</u>: Calculate the competitive ratio of the strategy in Question 1.2.

<u>Question 1.4</u>: What is the strategy that <u>maximize</u> the competitive ratio for this situation? What is the <u>largest</u> competitive ratio?

From next question, assume that the lump sum amount is X Million Yen, if a person receive it at 60, and the yearly pension is Y Million Yen, when X and Y is given as a known input.

<u>Question 1.5</u>: Devise a strategy that maximize the competitive ratio for this situation. What is the largest competitive ratio?

<u>Question 1.6</u>: Discuss why the idea from Gotenshita (rent-vs-buy) problem discussed in the class does not work for this problem.

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

Final Problem 2

In this problem, we will continue working on the pension system in Problem 1. However, instead of considering the competitive ratio, we will consider the expected income we can receive from the pension system in this problem. By the genome information, we know the probability that we will die at all specific ages.

In Question 2.1 – 2.4, let try to find a strategy that maximize the expected income by the optimization model.

Question 2.1: State the input of your optimization model using mathematical formulations.

Question 2.2: State the output of your optimization model using mathematical formulations.

Question 2.3: State the constraint of your optimization model using mathematical formulations.

Question 2.4: State the objective function of your optimization model using mathematical formulations.

Question 2.5: Devise an efficient algorithm for solving your optimization model.

Question 2.6: Assume that the lump sum amount is 80 Million Yen, when a person receive it at 60, and the yearly pension is 8 Million Yen. The probability that the person lives until x is $1/40$ for all $x \in \{60, 61, \ldots, 99\}$. What is the best strategy for this situation? What is the expected income for that best strategy?

Question 2.7: The best strategy when we consider competitive ratio is different from the best strategy when we consider expected income. Discuss when we should use the first strategy, and when we should use the second strategy.

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

Final Problem 3

In this problem, we will find the relationship between the k-center problem and the k-median problem. The k-median problem is given as follows:

<u>Input</u>:	Number of points n (Set of points $= \{1, \ldots, n\}$), Number of ward offices K d_{ij}: distance between i and j (Assumption: $d_{ij} \leq d_{ik} + d_{kj}$ for all i, j, k)
<u>Output</u>:	A set of points for ward offices $S \subseteq \{1, \ldots, n\}$
<u>Constraint</u>:	$\|S\| = K$
<u>Objective Function</u>:	$D_i^{(S)} \coloneqq \min_{j \in S} d_{ij}$: distance from point i to closest ward office minimize $\sum_{i=1}^{n} D_i^{(S)}$

Define the following notations:

$SOL_{median}(S)$: Value we get from applying the objective function of k-median problem to the output S.
$SOL_{center}(S)$: Value we get from applying the objective function of k-center problem to the output S.
OPT_{median}: Optimal value for k-median problem
OPT_{center}: Optimal value for k-center problem

<u>Question 3.1</u>: Discuss why, for any output S, $SOL_{center}(S) \leq SOL_{median}(S)$.
Hint: $\max\{a_1, \ldots, a_n\} \leq \sum_{i=1}^{n} a_i$

<u>Question 3.2</u>: Discuss why, for any output $S, SOL_{median}(S) \leq n \cdot SOL_{center}(S)$.
Hint: $\sum_{i=1}^{n} a_i \leq n \cdot \max\{a_1, \ldots, a_n\}$

<u>Question 3.3</u>: Discuss why $OPT_{median} \leq n \cdot OPT_{center}$.

From next question, suppose that $5n$-approximation k-center problem is NP-hard (although, we all know that it is not).

<u>Question 3.4</u>: Write a program for solving $5n$-approximation k-center problem based on the fact that a program for 5-approximation k-median problem is available in a library.

```
Sets 5ApproxKCenter(float[][] d);
Sets 5nApproxKMedian(float[][] d){
      \\Your code go here
}
```

<u>Question 3.5</u>: Discuss the reason why your code in Question 3.4 can solve $5n$-approximation k-median problem.

<u>Question 3.6</u>: Based on the (non-realistic) assumption that $5n$-approximation k-center problem is NP-hard, discuss why a 5-approximation algorithm for k-median is unlikely to be found.