Reminder: You may work in groups and use outside sources. But, you must write up solutions in your own words and properly reference your sources for each problem. This includes listing your collaborators and properly citing any sources you use. Solutions to each problem must be electronically typeset in separate documents and submitted online via Canvas.

Problem 8-1. Practice Probability

(a) KT Chapter 13, Problem 2.
(b) KT Chapter 13, Problem 5.
   
   Hint 1: It is proposal C. First write down the formula for the cost for town \( T_i \) using proposal \( C \) in terms of \( k, n \) and \( i \).
   
   Hint 2: In order to calculate \( E[X_i] \), you want to divide \( X_i \) into a bunch of random variables. In particular, say \( Y_{i,j} \) is an indicator RV which is 1 if town \( i \) pays for link \( e_j \) and 0 otherwise. Notice that \( Y_{i,j} = 1 \) if \( T_j \) arrives before all other towns larger than \( j \) and is 0 otherwise. So you can calculate \( E[Y_{i,j}] \) and then write \( X_i \) in terms of \( Y_{i,j} \)'s.

Problem 8-2. Making Binary Search Dynamic

Binary search of a sorted array takes logarithmic search time, but the time to insert a new element is linear in the size of the array. We can improve the time for insertion by keeping several sorted arrays.

Specifically, suppose that we wish to support SEARCH and INSERT on a set of \( n \) elements. Let \( k = \log(n + 1) \), and let the binary representation of \( n \) be \( n_{k-1}, n_{k-2}, \ldots, n_0 \). We have \( k \) sorted arrays \( A_0, A_1, \ldots, A_{k-1} \), where for \( i = 0, 1, \ldots, k - 1 \), the length of array \( A_i \) is \( 2^i \). Each array is either full or empty, depending on whether \( n_i = 1 \) or \( n_i = 0 \), respectively. The total number of elements held in all \( k \) arrays is therefore \( n \). For instance, if \( n = 26 \) with binary representation 11010, \( A_1, A_3 \) and \( A_4 \) will be full and \( A_0 \) and \( A_2 \) will be empty.

Although each individual array is sorted, there is no particular relationship between elements in different arrays.

(a) Describe how to perform the SEARCH operation for this data structure. Analyze its worst-case running time.
(b) Describe how to insert a new element into this data structure. Analyze its worst-case and amortized running times. Remember that your insert procedure should maintain the data structure invariants — if there are $n$ total elements in the data structure, you must have $A_0, A_1, ..., A_{k-1}$ sorted arrays and those corresponding to the 1 in the bit representation of $n$ should be completely full and the rest should be empty.

*Hint:* Your worst case cost should be $O(n)$ and the amortized cost should be $O(\lg n)$.

**Problem 8-3. Bidding Strategies**

(a) *KT* Chapter 13, Problem 9. I will give you the strategy — watch the first $n/2$ bids without accepting any of them. After that, accept any bid that is higher than those seen so far. If no such bid appears, accept the last bid. Argue that this strategy gets the biggest bid with probability at least $1/4$.

(b) *KT* Chapter 13, Problem 10.

*Hint:* Define indicator random variables $X_i = 1$ if the $i$th bid updates $b^*$. 

2