DO NOT POST YOUR CODE IN PIAZZA.

Read all the questions carefully. You can ask for clarification in piazza, but not for answers. You can search over the internet to learn, but must not ask questions to any online forum except for piazza.

All your programs will be checked by a program. Therefore, please strictly follow the instructions. The questions have been designed from scratch. Report any error about the question in Piazza - if you find any.

All the codes will be checked with a plagiarism checker, and any case of plagiarism, if detected, will be reported to the department.

If your overall score is below 30% in this assignment, then you might be considered for some manual partial marking, but that will not happen before the midterm.

A. Network Delay

Assume that a large database is being transferred from a source S to a destination D. The transfer is being done through many routes in the form of small data packets. The destination puts the data packets on a stack in the order they arrive. Since different paths have different time requirement, a set of packets p_1, p_2, p_3, p_4, p_5 may be collected at D in the order $S = (p_3, p_1, p_5, p_4, p_2)$.

A network performance metric t for this data transfer is computed by taking for every pair of packets $p_i, p_j \in S$, where p_i arrives before p_j, the maximum of $(i - j)$.

In the above example, $t = 3$, which is determined by p_5 and p_2

Input:

The input is in ‘network.txt’. Each line contains a list of n numbers ($1 \leq n \leq 1,000$) representing the packet ordering at D.

Sample input:

```
3,1,5,4,2
10,3,200
```

Output:

Each line of the output contains one number, which is the performance metric for the corresponding input.

Sample output:

```
3
7
```

Instructions:

Edit the given code Network.java to complete the assignment. Do not edit the part which is READ ONLY. Only submit Network.java in Moodle. To obtain a full mark for this question, you must try to achieve an $O(n \log n)$ time algorithm.

B. Mass Production of Quantum Computer

This is expensive, so before the production - we want to find the best possible design.

The specification says that the computer has X 'data bus' — you can think this as X parallel and disjoint horizontal lines lying on the lines $y = k$, where $1 \leq k \leq X$.

The computer needs to connect each pair of buses. The connections can only be vertical. The cost of connecting a pair of buses is the number of other buses that it crosses. The task is to find the minimum total cost for arranging the buses.

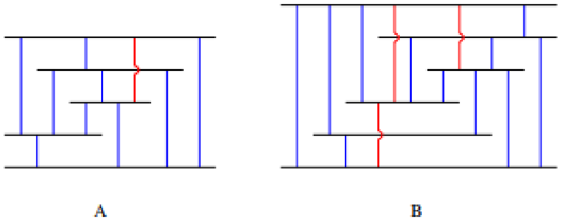


Figure A is showing a configuration with cost 1, and this is the minimum possible when $X = 5$. Note that there can be exactly 10 possible connections for 5 buses and we can make 9 of these connections without crossing. The one shown in red is the one where there is no way we can avoid a crossing.

Figure B is showing a configuration with cost 3, where $X = 6$. I am not going to tell you whether this is the minimum possible - ask others in piazza.

Although in the above examples each connection crosses at most one bus, for larger values of X, a minimum cost solution may contain a connection with two or more buses.

Input:

The input is in 'quantum.txt'. Each line contains a number X, representing the number of data buses.

Sample input:

```
1
4
5
```

Output:

Each line of the output contains one number, which is the cost for the corresponding input.

Sample output:

```
0
0
1
```

Instructions:

Edit the given code Quantum.java to complete the assignment. Do not edit the part which is READ ONLY. Only submit Quantum.java in Moodle. To obtain a full mark for this question, you must try to achieve answers for larger values of X.

Reward:

If you can find a correct answer for $X = 8$, then you get a reward. You can submit for a reward anytime before November 30.

C. Just to Sharpen the Brain

Let's define a class of graphs called 'Recursive Network' consisting of $n \geq 4$ nodes. The base graph looks like the top left graph in the following figure. The subsequent graphs are produced by inserting new nodes in clockwise order.

Given a recursive network, we want to find a crossing free layout for the graph minimizing rectangular area of the layout. The term 'crossing free' means no two links cross in the layout.

The bottom figure shows an example crossing free layout that minimizes the area when $n = 7$. The rectangular area is (width × height) of the layout. In this example, the area is $3 \times 6 = 18$ unit square, which is the best possible.

Instructions:

Write your answer and upload a single pdf in moodle. Your answer must consist the followings.

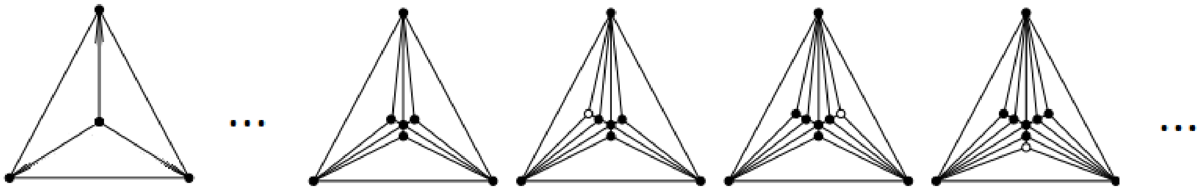
(a) A recurrence relation for solving the problem using dynamic programming.

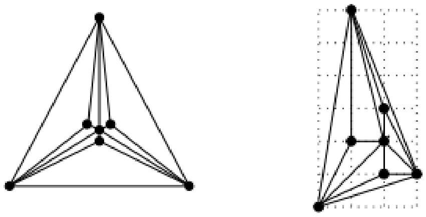
(b) A pseudocode for solving the problem using your recurrence relation.

(c) A brief argument of why you think your algorithm will produce a correct answer.

(d) The description of the dynamic programming table (or data structure, if any) that you will be using.

(e) Time complexity analysis of your dynamic programming.





To get a full mark, your algorithm must have a time complexity, which is a polynomial in n.

Reward:

If you can find a correct answer with a running time around $O(n^9)$, then you will get a bonus. Here n is the number of nodes. You can submit for a reward anytime before November 30.