Part A (35 points)

Problem 1

You are participating in the design of a new stepping stones challenge for the remake of the cult Japanese TV show "Takeshi's Castle".

The challenge involves a team of two people tied with a rope that need to walk over a sequence of stepping stones. The first teammate is allowed to go over the stepping stones that are painted red $\{r_0, \ldots, r_n\}$, while the second teammate is allowed to go over the stepping stones that are painted blue $\{b_0, \ldots, b_m\}$. For the team to win, the two players have to walk over all the stepping stones in the corresponding sequence while they are connected with the rope. The teammates are not allowed to backtrack. At each point in time, either one of the players can jump from her current stone to the next one and the other one stays at his current stone, or both of them jump simultaneously from their current stones to the next ones. Such a jump is obviously feasible only if the distances between the two players before and after the jump are less than the length of the rope connecting them.

The TV show producer has already built the two sequences of red and blue stepping stones. Given the coordinates of the stepping stones, your task is to select the minimum length of the rope for which it is possible for the two players to win the game. This is what will make the show entertaining for the audience!

Provide an algorithm for this computation and argue its running time.

This problem is the same as the edit distance but instead of two words we have from r_0, r_n and b_0, b_n. Similar to that problem from class, we must solve a prefix of r_0, r_j and b_0, b_j. There are three sub problems: $r[j]$ and a gap, $b[j]$ and a gap, or $r[j], b[j]$ meaning they are the same or different and nothing needs to be added. We can use what is in the DPV book, $E(i,j) = min\{1 + E(i-1,j), 1 + E(i,j-1), diff(i,j) + E(i-1,j-1)\}$. where $diff$ is the difference, 0 if they are the same and 1 if they are different. Rather than finding the edit distance for words, we find the edit difference between the movements. Our max number, n, is the length of the rope. If we need to add a gap, we increase by one. If they jump together, then we do not. The largest this distance can be is the length of the rope. Because of this, the total running time to compute this is $O(nm)$.

Part B (35 points)

Problem 2

A. **You have a collection of n distinct chopsticks of length $l_1, \ldots, l_n$. Any two of them can be paired for use if the length of them differ at most k. How can you easily pair as many of the chopsticks as possible? Describe a greedy algorithm of time complexity $O(n \log n)$ to solve this problem and prove the correctness of your algorithm.**

B. **Consider now a variant of the above problem. You can still only pair chopsticks that differ at most k in length. But now a value w_i is also associated with each individual chopstick. You want to maximize the sum of the values of the chopsticks that have been paired.**

For example, suppose you have 7 chopsticks of length $5, 2, 3, 11, 9, 12, 16$ and corresponding values $1, 1, 2, 5, 3, 3, 10$. You are allowed to pair chopsticks that differ by at most 3 units in length. Then one of the optimal solutions here is $\{(2,3), (9,11)\}$ of optimal value $1 + 2 + 3 + 5 = 11$.

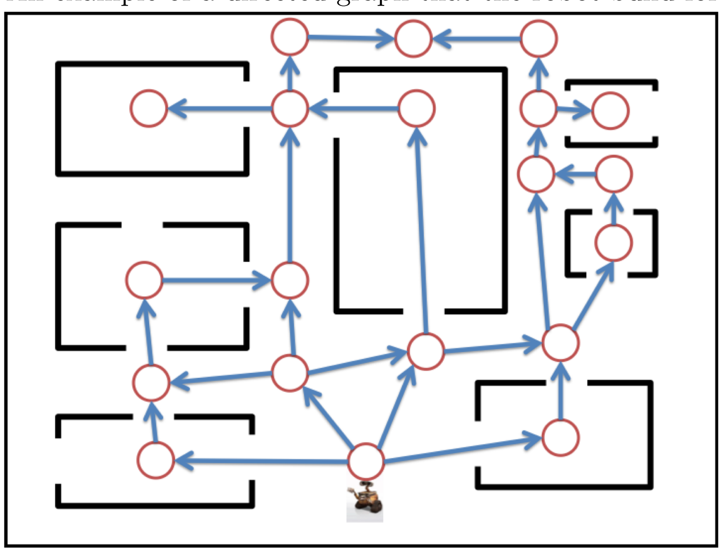
How can you pair the chopsticks so as to maximize the value? Describe a dynamic programming algorithm of time complexity $O(n^2)$ to solve this problem. Can you do better than $O(n^2)$?

Part C (20 points)

Problem 3

In the robotics lab the new robot has just arrived. The robot has the ability to construct a topological map of the environment, such as the graph shown in Figure 1. The robot is allowed to move only forward along the directions of the edges on the topological map. Moreover, the graph is being constructed in such a way that will prevent the robot to execute loops, i.e., the robot is not able to visit a node that it has already visited.

Figure 1: An example of a directed graph that the robot build for this map.



A. **Given a start location for your robot and a target location, provide an efficient algorithm that will return all the possible paths from the start to the target. What is the running time for your algorithm?**

The breadth-first search algorithm will return all possible paths between two points in a directed acyclic graph.

```
procedure all-paths-dag(x, y, G)
  Input:  Distinct vertices x and y in the directed-acyclic graph G
  Output: All possible paths from the the start to the target
```

B. **You want to check if the topological map provides enough information for your robot to be able to visit all the rooms so as to clean them. Provide an efficient algorithm that will be able to check if there is a path for the robot on the graph that can visit all the rooms (i.e., nodes on the graph).**

Part D (20 points)

Problem 4

You are preparing a banquet where the guests are government officials from many different countries. In order to avoid unnecessary troubles, you are asked to check the list of international conflicts in the last ten years. Then, you will assign the guests to two tables, such that in each table, any two guests are not from countries that had conflicts in the last ten years.

Provide an efficient algorithm that determines whether it is possible to make such an assignment. If it is possible to do so, the algorithm should return the assignment of these two tables. What is the running time?