1. Because of the recent droughts, N proposals have been made to dam the Murray river. The i^{th} proposal asks to place a dam x_i meters from the head of the river (i.e., from the source of the river) and requires that there is not another dam within r_i metres (upstream or downstream). What is the largest number of dams that can be built? You may assume that $x_i < x_{i+1}$.

   Hint: Consider the following subproblems $P(i)$ for all i such that $1 \leq i \leq n$:

   $P(i)$: find a subset of the set $S_i = \{x_1, \ldots, x_i\}$ with the largest possible number of elements indicating positions where to build the dams so that the conditions are satisfied but so that position x_i is included.

   This makes recursion easy: if for some j position x_j satisfies $x_j < x_i - r_i$, then this is also true for all $k < j$. Similarly, for all k such that $k < j < i$ and which satisfy $x_k + r_k < x_j$ then also $x_k + r_k < x_i$. Thus, to obtain an optimal solution for S_i it is enough to look for all j such that $x_j + r_j < x_i$ and $x_i - r_i > x_j$ and now you should be able to do a recursion. IMPORTANT: After you solve all these problem, optimal solution to the original problem need not be the solution for problem $P(n)$; now you have to pick the largest among optimal solutions for $P(1), \ldots, P(n)$.

2. We are given a checkerboard which has 4 rows and n columns, and has an integer written in each square. We are also given a set of $2n$ pebbles, and we want to place some or all of these on the checkerboard (each pebble can be placed on exactly one square) so as to maximize the sum of the integers in the squares that are covered by pebbles. There is one constraint: for a placement of pebbles to be legal, no two of them can be on horizontally or vertically adjacent squares (diagonal adjacency is fine).

   (a) Determine the number of legal *patterns* that can occur in any column (in isolation, ignoring the pebbles in adjacent columns) and describe these patterns.

   Call two patterns *compatible* if they can be placed on adjacent columns to form a legal placement. Let us consider sub-problems consisting of the first k columns $1 \leq k \leq n$. Each sub-problem can be assigned a type, which is the pattern occurring in the last column.

   (b) Using the notions of compatibility and type, give an O(n)-time algorithm for computing an optimal placement.

   Hint: Finding all pairs of compatible patterns should be easy: for every possible column c of size 4 squares which does not contain (vertically) adjacent pebbles look for all patterns which also do not contain vertically adjacent pebbles and, in addition, do not contain pebbles in the rows already containing a pebble in column c. Now for each i and each legitimate column c solve the problem of finding optimal solution for checkerboard with i columns, but such that the last, i^{th} column is c. Remember, a column with no pebbles at all is compatible with every other column. After you solve the problems of size n for each possible ending columns, pick among such solutions one with the largest score.

3. Skiers go fastest with skis whose length is about their height. Your team consists of n members, with heights $h_1, h_2, \ldots, h_n$. Your team gets a delivery of $m \geq n$ pairs of skis, with lengths $l_1, l_2, \ldots, l_m$. Your goal is to write an algorithm to assign to each skier one pair of skis to minimize the sum of the absolute differences between the height h_i of the skier and the length of the corresponding ski he got, i.e., to minimize

$$\sum_{1 \leq i \leq n} |h_i - l_{j(i)}|$$

where $l_{j(i)}$ is the length of the ski assigned to the skier of height h_i.

Hint: Order all skiers S_i, $1 \le i \le n$ by increasing height $h(S_i)$ and all skis s_j, $1 \le j \le m$, by increasing length $l(s_j)$. Now notice that if an assignment is optimal and $h(S_i) < h(S_j)$ then the skis assigned to skier S_i are shorter than the skis assigned to skier S_j, otherwise you could swap their skis without increasing the sum of the absolute values of the differences between the heights of skiers and length of skis. Consider now the following subproblems $P(i, j)$ for all i and j satisfying $1 \le i \le n$ and $i \le j \le m$: "Find optimal assignment of the first i skiers to chose from the first j skis." If $j = i$ then there is only one assignment that assigns skis according to the skier's height. If $j > i$ then there are two choices: either you assign to the i^{th} skier skis j or you do not. It should now be easy now to do a recursion.

4. You know that $n+2$ spies S, $s_1, s_2, \ldots, s_n$ and T are communicating through certain number of communication channels; in fact, for each i and each j you know if there is a channel through which spy s_i can send a secret message to spy s_j or if there is no such a channel (i.e., you know what the graph with spies as vertices and communication channels as edges looks like).

   (a) *Your task is to design an algorithm which finds the fewest number of channels which you need to compromise (for example, by placing a listening device on that channel) so that spy S cannot send a message to spy T through a sequence of intermediary spies without the message being passed through at least one compromised channel.*

   (b) Assume now that you cannot compromise channels because they are encrypted, so the only thing you can do is bribe some of the spies. Design an algorithm which finds the smallest number of spies which you need to bribe so that S cannot send a message to T without the message going through at least one of the bribed spies as an intermediary.

   Hint: a) should be a straightforward max flow. What should you assign as the capacity of edges so that the min cut equals to the number of edges crossing the cut? For b) you might want to use the same trick we used to find max flow when not only edges but also some of the vertices had a limited capacity: duplicate each spy vertex and connect the two vertices with an edge of a unit capacity. What should be the capacities of all other edges? Note that your algorithm might also output "No solution - S can communicate with T by a direct channel."

5. You are given a flow network G with $n > 4$ vertices. Besides the source s and the sink t, you are also given two other special vertices u and v belonging to G. Describe an algorithm which finds a cut of the smallest possible capacity among all cuts in which vertex u is at the same side of the cut as the source s and vertex v is at the same side as sink t.

   Hint: it is enough to ad two edges, but students often prefer to add a super source and a super sink