

Q1

As the question mentioned, we first construct a bipartite graph with all the cities as vertices on the left side and all cities as vertices on the right side. Cities on the left side represent populations of the corresponding cities, cities on the right side represent the set of pods in the corresponding cities.

If there exists a way from city C_i on the left side to city C_j in cities on the right side that cost less than X days (i.e. $t(i, j) \leq X$), whatever a direct road from C_i to C_j or a sequence of intermediate cities connected by direct roads from C_i to C_j , connect C_i to C_j as one direction edge with infinite capacity. Thus, we add a super source S and a super sink T and connect S with all the cities C_i in cities on the left side with edges of inhabitant and with capacity of the population. Also, connect all cities on the right side with T as edges with capacity of the number of pods.

We now run the *Edmons-Karp* algorithm to find the maximal flow through such a network, which is the largest number of invaders the Earth will have to deal with.

Q2

As the question mentioned, make a bipartite graph with n vertices on the left side representing number n row on the chessboard and n vertices on the right side representing number n column on the chessboard. So edges from left side r to right side c ($1 \leq r, c \leq n$) corresponding to the cell (r, c) on the chessboard. Introduce a super source S and a super sink T . And connect S to each vertices on the left sides by directed edge with capacity equal to 1, also, connect each vertices on the right side by directed edge to T with capacity equal to 1, because no two black rooks are in the same row or in the same column. For the white bishops (a_i, b_i) , we have a set M which contains all possible points lies on the diagonal of (a_i, b_i) , ($1 \leq a_i, b_i \leq n$, $1 \leq i \leq k$). If there exists a cell $(r, c) \notin M$, connect r to c by directed edge. After that, just use max flow algorithm and find the max flow in such a network, that is the largest number of black rooks we can place on the chessboard.

Q3

As the question mentioned, we should to find the minimum cost to disconnect Computer 1 to Computer N , which is a minimal cut problem.

First, the source is Computer 1, and the sink is Computer N . Now we construct a flow network as a directed graph where the computers are the vertices of the graph and each edge is represented by two directed edges of opposite orientation and each of capacity equal to the cost of removing link from Computer i to Computer j or from Computer j to Computer i . Now, we run *Edmons-Karp* algorithm to find the maximal flow through such a network. After the algorithm has converged, we construct the last residual network flow and look at all the vertices to which there is a path from source Computer 1. This will define a minimal cut, so we look at all the edges crossing such a minimal cut. The sum of edges which in the forward direction determines the minimum cost that needed to disconnect from Computer 1 to Computer N .