Assignment 5 solutions

1. Today was just a regular day for everyone in Krypton until a news flashed that a meteor is going to destroy Krypton in X days. Krypton has N cities, some of which are connected by bidirectional roads. You are given a road map of Krypton; for every two cities C_i and C_j which are connected by a (direct) road from C_i straight to C_j you are given the value $t(i,j)$ which is the number of days to travel from city C_i to city C_j. (You can of course also go from a city C_m to city C_k without a direct road from C_m to C_k by going through a sequence of intermediate cities connected by direct roads.) In each city C_i the Krypton Government built q_i pods to carry inhabitants in case of any calamity, which will transport them to Earth. City C_i has population p_i. As soon as the people hear this news they try to save themselves by acquiring these pods either at their own city or in other city before the meteor destroys everything. Note that a pod can carry only one person. Find the largest number of invaders the Earth will have to deal with. (20 pts)

   Solution: First apply Floyd Warshall's algorithm to the weighted graph with cities as its vertices and edges representing roads, with each edge having a weight equal to the time $t(i,j)$ needed to traverse it. In this way we find the shortest distance from every node to any other node and determine which of these distances are less than X. We now construct a bipartite graph; on the left there will be N nodes l_i, one for each city c_i representing population of city c_i; on the right there will be N nodes r_j, again one for each city c_j, but this time representing the set of pods available in city c_j. We now connect each left node l_i with all right nodes r_j which satisfy that the shortest path from city c_i to city c_j is less than X days in total (note that l_i is always connected with r_i because it takes 0 days to travel from c_i to c_i). The capacities of such connecting edges can be set to either infinity or to the total number of inhabitants in city c_i. We now add a super-source and connect it with all nodes l_i with edges of capacity equal to the number of inhabitants in city c_i; we also add a super-sink and connect every node r_j with the super-sink by an edge of capacity equal to the number of pods in city c_j. We now find a max flow in such a flow network; such max flow represents the total number of Kryptonians who will be able to escape to Earth.

   The complexity of such an algorithm is equal to the sum of complexities of the Floyd Warshall which is $O(n^3)$ plus the complexity of the max flow algorithm, and the fastest one is also $O(n^3)$. Thus, the total complexity of the algorithm is $O(n^3)$.

2. You are given an $n \times n$ chess board with k white bishops on the board at the given cells (a_i, b_i), $(1 \leq a_i, b_i \leq n$, $1 \leq i \leq k)$. You have to determine the largest number of black rooks you can place on the board so that no two rooks are in the same row or in the same column or are under the attack of any of the k bishops (recall that bishops go diagonally).(20 pts)

   Solution: To solve this problem we construct a bipartite graph with n left vertices r_i representing n rows of the board and n right vertices c_j representing n columns of the board. We construct edges in such a graph so that vertex r_i is connected with a vertex r_j just in case the cell (i, j) on the board is not under attack of any of the bishops. We add a super source s and connect it with all vertices r_i with edges of capacity 1; we also add a super sink and connect all vertices c_j also with edges of capacity 1. The maximal number of rooks that meet the conditions is equal to the max flow in this flow network, with rooks placed in the cells corresponding to the occupied edges from r_i to c_j.

3. There are N computers in a network, labelled $\{1, 2, 3, \ldots, N\}$. There are M one-directional links which connect pairs of computers. Computer 1 is trying to send a virus to computer N. This can happen as long as there is a path of links from computer 1 to computer N. To prevent this, you've decided to remove some of the links from the network so that the two computers are no longer connected. For each link, you've calculated the cost of removing it. What is the minimum total cost to disconnect the computers as required, and which edges should be removed to achieve this minimum cost? (20 pts)

   Solution: We model this as a max flow problem. We let computer 1 be the source, computer N be the sink. All the edges will have capacity equal to the cost of removal. This then becomes a s-t min cut problem. Removing the min cut edges guarantees that it will completely block the virus from entering computer N, and will also minimise the total cost of the edges removed as well.