

COMP3121 Homework Q1

Arth Sanskar Patel
z5228942

August 6, 2020

1 Answer

Here we have cities numbered from 1 to n . Each city C_i has a population p_i and q_i number of pods that can carry people to earth. Cities are connected by roads and it takes different amount of time to go from one city to another. In X days krypton will be destroyed. This can be easily solved using a bipartite graph. For this question, first, let's see from all the list of cities, which city can we reach in given X days. We can simply use normal BFS algorithm with a minor change inside the while loop. We can keep a counter of weights of each edge we traverse (the number of days) and when that counter equals X we break that loop and start again. This way for all the cities (vertices) we will only have the cities it can reach in X days. Next is making our bipartite graph. On the left hand side we have our cities listed from C_1 upto C_n . All these cities are connected to a super source with the edge having capacities equal to the population of the cities. On the right hand side let's again have the cities but let's call them A_1 to A_n (note here that A_1, \dots, A_n represents the cities only but I chose a different variable name to avoid confusion). Connect the right hand side to a super sink with each edge having capacity equal to the number of pods in the city which the vertex represents. For example edge from A_i to super sink will have a capacity of q_i . Now let's join the left and right of our graph. First off we connect all cities to their corresponding pods. C_1 to A_1 , C_2 to A_2 and so on. This is done because people in any city can access the pods in their own city. Now let's go over the results of BFS and connect the cities which are accessible in a time of X days. So for example people of C_i can go to any city C_j we will connect C_i on the left to A_j on the right. The edges connecting C 's to A 's will have an infinite capacity since any number of people can go from one city to another. Here we can also keep these edges with capacity equal to the population of origin cities but since we already took that in account when connecting to the super source, we need not do that again.

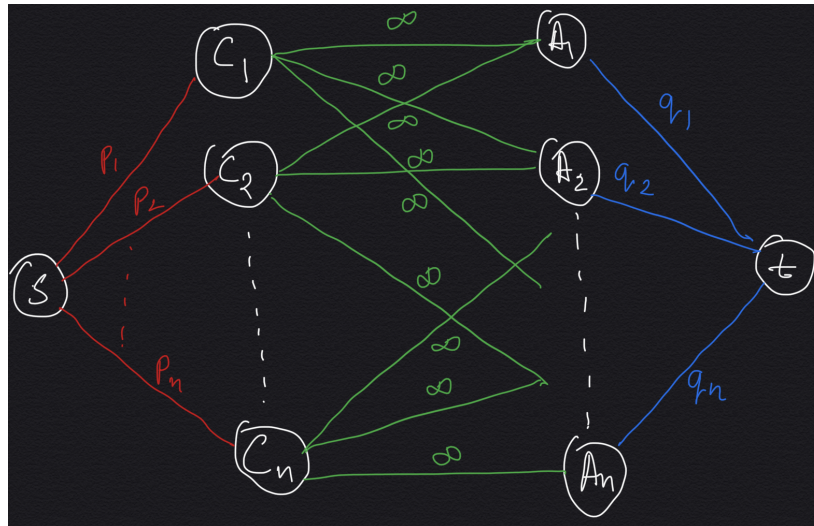


Figure 1: Bipartite Graph

Finally call Ford-Fulkerson Algorithm on this graph to get the maximum flow. The total flows going towards the super sink will be the amount of invaders that earth has to deal with. Reference for this question is taken from Tutorial 5 question 9, <https://www.youtube.com/watch?v=ar6x7dHfGHA&t=231s> and <https://www.hackerearth.com/practice/algorithms/graphs/maximum-flow/practice-problems/algorithm/doomsday/description/>.