Matthew Soicher
z5160737

Question 1

We begin the problem by first finding for every city C_i, the cities which citizens can reach in search of a pod within X days of the meteor hit, i.e. $t(i, j) < X$.

This will be done by using a standard DFS algorithm.

Then, to maximise the number of Kryptonite's who are able to safely flee Krypto, we construct a bipartite graph whereby each vertex represents any city C_i with a connection to any city C_j denoting the ability for citizens to travel to that city in search of a pod within the time constraint X, as found in our search beforehand.

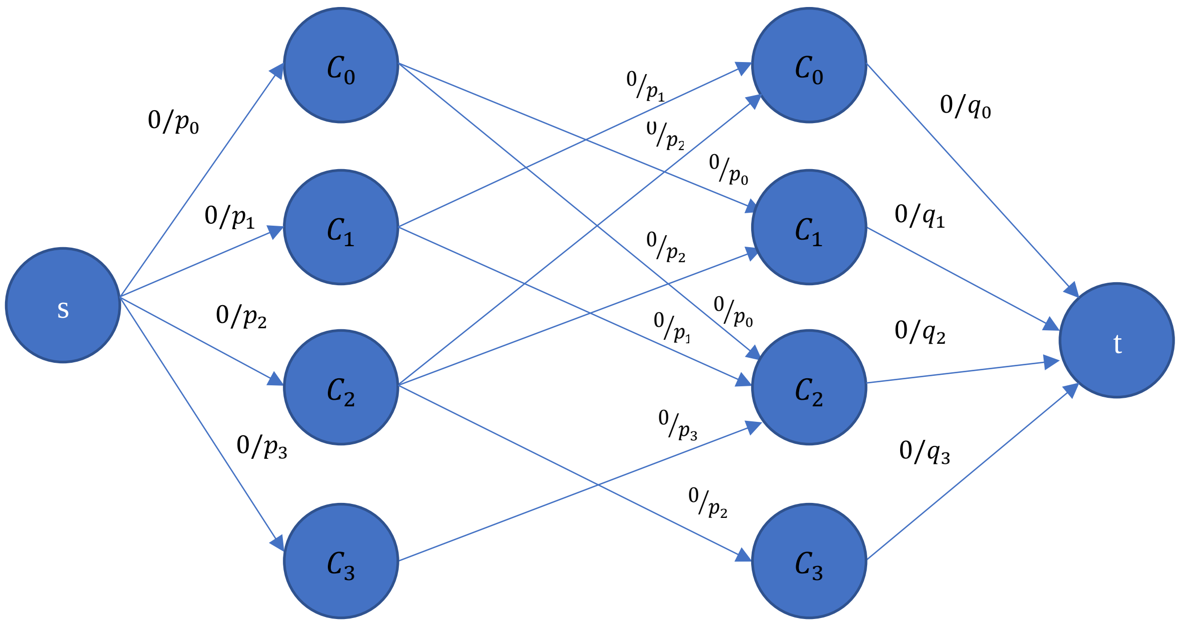
For every edge connecting from the source node to any vertex C_i on the left-hand side, its weight is given by that city's population p_i. Similarly, the edges connecting all vertices $C_{0...N}$ to reachable cities will be given a weight of that outgoing city's population, C_i since that is the maximum flow that can go through that path,

Alternatively, for connections from the right-hand side's vertex's C_i to the sink node, its weight is given by the city's pod population q_i.

For example, say we have cities C_0, C_1, C_2, C_3 with respective populations p_0, p_1, p_2, p_3 and pod amounts q_0, q_1, q_2, q_3. Our search finds reachable cities in our time constraint like so:

- Citizens in C_0 can reach C_1, C_2
- Citizens in C_1 can reach C_0, C_2
- Citizens in C_2 can reach C_0, C_1, C_3
- Citizens in C_3 can reach C_2

As a result, this is how our bipartite graph as a representation of the flow network will look



Matthew Soicher
z5160737

Now, all that is required is to run the **Ford-Fulkerson Algorithm** on this flow network which will find the maximum flow (maximum number of citizens that can flee) for our problem.

The time complexity of the algorithm is $\boldsymbol{O(E \times f)}$ where E = number of edges in the graph and f is the maximum possible flow.