**Question 1**

We begin the problem by first finding for every city , the cities which citizens can reach in search of a pod within days of the meteor hit, i.e. .

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 with a connection to any city denoting the ability for citizens to travel to that city in search of a pod within the time constraint , as found in our search beforehand.

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

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

For example, say we have cities , , , with respective populations , , , and pod amounts ,. Our search finds reachable cities in our time constraint like so:

* Citizens in can reach,
* Citizens in can reach,
* Citizens in can reach,
* Citizens in can reach

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

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 where = number of edges in the graph and is the maximum possible flow.