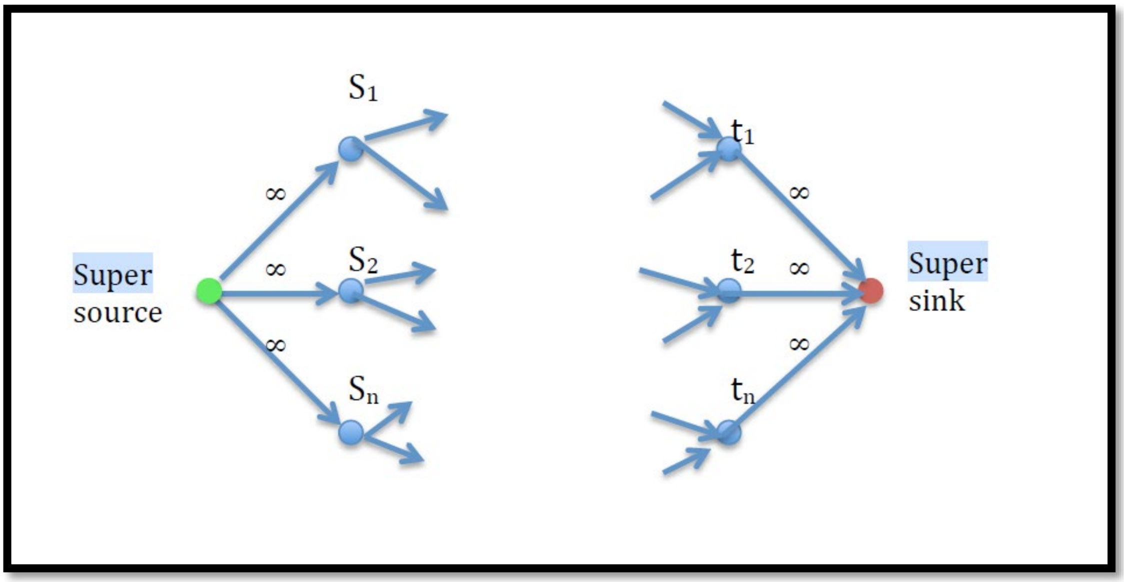
Dheeraj Satya Sushant Viswanadham
(z5204820)

COMP3121 Assignment 5 – Question 1

1) NOTE: The below solution is for the old question 1 before Aleks emailed us the new version on 01/08/2020, however, I believe my logic and reasoning still applies to this new version of the question.

We are required to find the largest number of "invaders" that Earth will have to deal with, or in other words, the maximum number of people that the pods in all cities on Krypton can carry to safety within *X* days, where it takes $t(i, j)$ days to travel from city *i* to city *j* having a direct road between them. We can recognise that this situation is similar to a Max flow problem, and we can solve it in a relative manner. We know that Krypton's inhabitants can escape Krypton if they are able to access a pod either in their city or another within *X* day(s) time from their own city. We can make a simple bipartite graph with the left side representing the population of the city and right side representing the respective pods in the city (similar to a source and sink situation). We can then add a super source connected to the source cities population and a super sink connected to the sink cities pods (which have edges of infinite capacities) to determine the maximum 'flow' that we can handle by taking the residual graph after accounting for the remaining flow.

See below for an overall visual diagram (taken from lecture notes) for what we are trying to achieve:



The problem to consider here is that since we only have *X* days to have the inhabitants escape Krypton, we would need to sort the cities in decreasing order with the largest number of pods and then out of these, the cities with the largest population in decreasing order, and then out of these, sort in decreasing order, the cities with the largest number of connections to other cities respectively, so that we can obtain the largest number of survivors in the limited time that we have. We will follow this order as if we instead started with sorting the cities with the largest population, we may not be able to obtain the maximum number of survivors as it ultimately depends on the number of pods that can take the survivors out of Krypton as if we sorted it out of population first, then the largest cities may not have enough pods and travelling to another city would not be an efficient use of the time. We sorted the cities with the greatest number of connections as that would ensure that out of the cities with the largest number of pods and population, the ones with the largest number of connections to other cities will be prioritised as being our sinks. This will have a time complexity of $O(n^4 lo\ g(n))$ using mergesort, as we have three for loops to go through where *n* is the same number of cities in all of them. Once we have our bipartite graph and attach our super source (on the left) and super sink (on the right), we can then assign our capacities to the roads (which will be the remaining number of pods in a city available after we have allocated the pods amongst the cities own inhabitants). We can then apply the Ford-Fulkerson algorithm to find the maximum number of pods that can be distributed to citizens of other cities (or their own if the city did not have enough) within $t(i, j)$ days which must be less than X days (as otherwise it will be too late). Additionally, we must make sure that if we are travelling from city *m* to city *k* where there is no direct road between them, then the time taken to travel from city *m* to any intermediary cities with the shortest path to city *k* is less than our *X* days time as otherwise it will be too late. We can apply Floyd-Warshall's algorithm in this case to find the shortest time in this case to travel from city *m* to city *k* which has a time complexity of $O(V^3)$ where V is the number of cities (or vertices) in our graph.

End of Solution