Given: N cities with trip time T to each other and their own pods and populations
Aim: Find how many pods can escape at maximum condition.

Prepossessing: Regard City Ci as the starting point use Dijkstra algorithm to find all possible cities that can be reached from Ci within T days trip time. Append all the found cities into list L. Do this for all the N cities, so that we will get N list L.

Setup: Construct a bipartite graph with N vertexes on left and right, At the leftmost there is a vertex which is the Super source S, then on the left hand side there are N vertexes which represent populations of each cities. At the rightmost is the Super Sink Si, on the right hand side there are N vertexes that represent number of pods of each cities. Connect with directed edge from S to all the vertexes on the left hand side, the capacity for each edge is equal to the connected city's(vertex) population. Then connect with directed edge from the vertexes on the right hand side to the rightmost, the capacity for each edge is equal to the connected city's(vertex) pods.

Solution: Connecting each city population with their own pods and connect the edge from left hand side to the cities (vertexes) on the right hand side that appeared in the corresponded List L. (For example: A city Ci, there are three cities Cj, Ck, Cm can be reached within X days then List Li will be [Cj,Ck,Cm], while connecting edges the Ci will only connect to Cj,Ck,Cm) The capacity for these edges are 0, because there are no restrictions about how many people pass the city, so that capacities are equal to 0. Now we finished graph. Then we find max flow in such a network using the Edmons-karp algorithm. Then build the last residual network and find the minimal cut. The result will be the total capacity of the minimal cut.