Question 2

There are k people living in a city, whose n suburbs and m roads can be represented by an unweighted directed graph.

Every person is either slow or fast. Every night,

- Each slow person can stay in the same suburb, or move along a road to an adjacent suburb.
- Each fast person can stay in the same suburb, or move to any other suburb in the entire city.

Over the last d days, you know how many people were located in each city. That is, for each day i and city j (where $1 \leq i \leq n$ and $1 \leq j \leq m$), you know $p_{i,j}$, the number of people located in city j on day i. You are guaranteed that $\sum_{j=1}^{m} p_{i,j} = k$ for all i.

2.1 [**10 marks**] Design an algorithm which runs in $O(d(n+m))$ time and determines whether there could possibly be at least one slow person.

Answer:

According to the settings in the topic, assume that each of the n suburbs acts as a node to form a network. Set a source node is linked to all nodes with the capacity of $p_{i,j}$. Set the capacity of all nodes to be 1 connected to the sink node.

According to the topic, the fast people can move quickly which also means that if the road had people, the people must be slow people. We can use Ford-Fulkerson algorithm to find the maximum flow. If the maximum flow is equal k, it means there are at least one slow people. If not, there is no slow people.

The time complexity of Ford-Fulkerson is $O(E * k)$, in this answer, $k = d$, the edge is equal $n + m$, the time complexity is $O(d(m + n))$.

2.2 [**10 marks**] Design an algorithm which runs in time polynomial in n, m, and d, and determines the minimum possible number of fast people.

Your answer here.