Homework 9

CS41, Spring 2023

Alina Palacios and Nina Zhuo

1. We make a bipartite graph where one set of n vertices represents all volunteers and another set of m vertices represents all residences. For every volunteer n_i, we draw an edge to each m residence that is less than a kilometer away such that $\sqrt{(x - x_i)^2 + (y - y_i)^2} \leq 1$.

   We know that each volunteer will spend 5 minutes at each residence they visit and we want the max flow $f*$ of our flow network to be the max number of residences visited, so each flow of 1 sent through the flow network will represent 5 minutes. This will also ensure that there will never be a case where two volunteers visit the same residence for a total of 5 minutes.

   We draw an edge from a dummy start vertex s to each volunteer n_i weighted on the amount of time volunteer n_i has committed t_i in minutes, adjusted according to our five minute modification such that each $s - n_i$ edge will be weighted on $12t_i$. Thus, each flow of 1 sent through the network is a potential 5 minutes any n_i volunteer could spend at any of the m residences.

   Finally, we draw an edge from each m to our end vertex t weighted on 1 to tally up the total number of residences the volunteers were able to visit.

   We run F-F in $O(|E|f*)$ where max flow $f*$ will be the max number of residences visited m and $|E| = O(nm)$ because at most, each n_i volunteer could visit m residences giving us $O(n^2m)$ total running time.

2. We will make a bipartite graph where there are two sets of vertices: the first set are all the boxes b_i and the other set of vertices are the boxes in which the boxes from the first set can fit into, b_j. We will make a source s and a sink t. First, we draw an edge from the s to each box in b_i, with the capacity being 1. Next, we draw and edge from each box in b_i to b_j based on the following equation:

   $$E = \{(i, j) : \text{box } i \text{ fits in box } j\} \cup (s, i), (i, t) : 1 \leq i \leq n$$

   Finally, we draw an edge from each b_j to our end vertex t weighted on 1 to determine the number of boxes b_i that can fit into b_j. We will count the number of boxes that cannot fit in another box by subtracting max flow $f*$ from total number of boxes which indicates the start of another trip.

   We run F-F in $O(|V|^3)$, where the max flow is the number of boxes in b_i that fit into the boxes in b_j. The runtime is $O(|V|^3)$ because we have $O(|V|^2$ since we can have boxes in b_i can fit into every b_j box. Then, we can set a flow of 1 from s $|V|$ number of times and that can potentially hit any of the b_i to b_j edges, thus resulting in $O(|V|^3)$.

3. We will make a graph where each vertex represents some coordinate (i, j) on the nxn grid with antiparallel edges running from each (i, j) to every $(i+1, j)$, $(i-1, j)$, $(i, j+1)$, and $(i, j-1)$ when those vertices exist. We will have a source vertex s connecting to each hiker's coordinate (i, j), and we will have a sink t with an edge from to each bottom square of the grid (i, n). We do not specify any capacity for the edges, but set the following vertex capacities:

   No Snake $= k$ because each hiker will visit the square at most once

   Asleep $= 2$ because at most two hikers can pass through without being bitten

   Awake $= 1$ because at most one hiker can pass through without being bitten

   Angry $= 0$ because the snake in this state will bite the hiker, so they cannot go through this vertex

   We run F-F to find max flow in $O(|E|f*)$-time. The max flow of the flow network $f*$ is k, since the source is going to k hikers. The number of edges is n^2, so time complexity is $O(kn^2)$.

4. We make a bipartite graph where one set of vertices represents each row i and another set of vertices represents each column j such that an edge from i to j represents a suitable site to potentially build a house. In this way, we will never have more than one house in each row and column such that each house has an unobstructed view of the north and the east. Each edge is given a capacity of 1 since only one house can be built at any given (i, j) on the grid.

   We draw an edge from vertex representing a column j (arbitrary; could also be the row i) to our end vertex t weighted on 1 such that the max flow $f*$ will never exceed n.

   We run F-F to find max flow in $O(|E|f*)$ time where $f*$ is the maximum number of houses that could be built n and $|E| = O(n^2)$ because for all potential n rows to place a house, we could have a potential suitable site in any n column, giving us total running time $O(n^3)$.