CSCI-GA.1170-001/002 Fundamental Algorithms December 3, 2015

Solutions to Problem 2 of Homework 10 (15 points)

Name: GOWTHAM GOLI (N17656180) *Due: Tuesday, December 1*

A fellow Moe and his buddy Joe live in a city $G = (V, E)$ which is an undirected graph on n vertices and m edges, given in the adjacency list form. Moe lives in a vertex a and owns a crazy dog Mimi, while Joe lives at a vertex b and owns a crazy dog Kiki. This Sunday Moe wants to take Mimi to a veterinarian clinic located at vertex c, while Joe wants to take Kiki to the dance competition located at a vertex d. One problem, though: the dogs hate each other, and if one of them smells the other, all hell breaks loose. Luckily, they can smell each other only if within distance at most 15 in G, and you know that $dist_G(a, b), dist_G(c, d) > 15$. Moe and Joe would like to start at the same time (with Moe and Mimi at a and Joe and Kiki at b) and get both dogs to their respective destinations c and d in the smallest number of *steps* t. A *step* consists of both dogs going to their respective neighboring vertices, or one dog going to a neighboring vertex, and the other dog staying put, barking at the pedestrians. Of course, such a step is possible only if the dogs stay within distance 16 or more both before and after the step.

Your job is to design an algorithm for Moe and Joe to compute t (and the optimal route), if the route exists, and analyze its running time.

(a) (5 pts) Using one or more runs of the BFS algorithm on G, fill in the matrix $OK[x, y]$, where $OK[x, y] = 1$, if it is OK for Mimi to be at vertex x and Kiki to be at vertex y at the same time, and $OK[x, y] = 0$ otherwise. How long did it take you to fill this matrix $OK[x, y]$?

Solution:

For every node $x \in V$ compute $BFS(x)$
and $\forall y \in V$, if $\delta(x, y) > 15$ set $OK[x, y] = 1$ and 0 otherwise.

In the above algorithm BFS is called on every node of G. Therefore the running time is $O(n(m + n)) = O(mn)$ □

(b) (5 pts) Design a graph $H = (V', E')$ whose vertex set consists of possible "ok configurations" for Mimi and Kiki, and whose edge set represents the possible single steps of your algorithm. Be sure to formally define V' and E' as functions of V and E and the matrix OK from part (a). How long (in the worst case) did it take you to create an adjacency list for H (not counting what you did in part (a))? What is the maximum $|V'|$ and $|E'|$?

Solution:

If for any $x, y \in V$, $OK[x, y] = 1$ then we create a single node for both x and y. Let this node be called xy. Therefore $V' = \{xy \text{ such that } OK[x, y] = 1 \text{ where } x, y \in V\}$. Therefore creating V' takes $O(n^2)$ time

Let x_1y_1 and x_2y_2 be any two vertices in V' then

- $(x_1y_1, x_2, y_2) \in E'$ if $OK[x_1, y_1] = 1$, $OK[x_2, y_2] = 1$ and $(x_1, x_2), (y_1, y_2) \in E$ i.e if Initially Mimi is at x_1 and Kiki is at y_1. Now each of them take one step, go to their respective neighbors x_2 and y_2 and they are still > 15 units apart from each other
- $(x_1y_1, x_2, y_2) \in E'$ if $OK[x_1, y_1] = 1$, $OK[x_1, y_2] = 1$ and $(y_1, y_2) \in E$ i.e if Initially Mimi is at x_1 and Kiki is at y_1. Now Mimi stays put at x_1 while Kiki goes to his neighboring vertex y_2 and they are still > 15 units apart from each other
- $(x_1y_1, x_2, y_1) \in E'$ if $OK[x_1, y_1] = 1$, $OK[x_2, y_1] = 1$ and $(x_1, x_2) \in E$ i.e if Initially Mimi is at x_1 and Kiki is at y_1. Now Kiki stays put at y_1 while Mimi to goes his neighboring vertex x_2 and they are still > 15 units apart from each other

Let $xy \in V'$. Now we need to analyze the number of outgoing edges from xy. Let m_x be the number of outgoing edges from x to it's neighboring vertices $x_1, \ldots, x_{m_x}$ and m_y be the number of outgoing edges from y to it's neighboring vertices $y_1, \ldots, y_{m_y}$ in G. Now for each $x_i \in \{x_1, \ldots, x_{m_x}\}$ and $y_j \in \{y_1, \ldots, y_{m_y}\}$, we check if $OK[x_i, y_j]$ is 1 and add an edge from xy to x_iy_j. Therefore the total time taken to add the number of outgoing edges from xy is m_xm_y.

Let $x_1, \ldots, x_n \in V$ and $y_1, \ldots, y_n \in V$ be all the vertices of G. In the worst case if $OK[x_1, y_i] = 1$ for each $i = 1, \ldots, n$ then the time taken to add all the outgoing edges from each vertex x_1y_i is $m_{x_1}(m_{y_1} + \ldots + m_{y_n}) = m_{x_1}m$. Similarly we can analyse for all $x_1, \ldots, x_n \in V$, Therefore the total time taken to create E' is $m(m_{x_1} + \ldots + m_{x_n}) = m^2$

Therefore the total time to create the adjacency list of H is the time to create V' and the time taken to $E' = O(m^2 + n^2)$.

In the worst case when all the vertices are more than 15 units from each other the entire OK matrix will be 1. Therefore $|V|_{max} = n^2$ and using the above analysis of how outgoing edges are created it is easy to see that $|E|_{max} = m^2$ □

(c) (5 pts) Describe the original problem as a shortest path computation on H. Finally, solve the original problem, and help Moe and Joe. Analyze the overall running time of your algorithm, as a function of n and m.

Solution:

All the vertices and edges of H are of OK configurations. So simply call BFS on ab and get the shortest distance from ab to cd and return $\delta(ab, cd)$. Running time will be $O(|V'| + |E'|) = O(m^2 + n^2)$ □