Homework 9

CS41, Spring 2023 | Due Sunday, April 9

Solve the following problems using the Ford-Fulkerson (or Edmonds-Karp) algorithm. Briefly explain why each solution is correct and achieves the stated running time.

1. Suppose you are in charge of deploying volunteers for door-to-door political canvassing in a city. There are m residences, and you have coordinates (x, y) for each one. Canvassing will take each volunteer five minutes per residence; you do not need to consider transit time. For each residence, these five minutes cannot be split between multiple volunteers.

   There are n volunteers, and for each volunteer i, you have the positive integer $t_i \leq 12$ of hours they can commit to canvassing and the coordinates (x_i, y_i) of their home address. You only want to assign volunteers to residences that are within one kilometer of their own home, i.e., where $\sqrt{(x - x_i)^2 + (y - y_i)^2} \leq 1$.

   Design a $O(n^2 m)$-time algorithm to find the maximum number of residences that your group of volunteers can canvass.

2. The box nesting problem from Lab 9 (which I don't think anyone solved during lab) can be reformulated as partitioning the vertex set of a given DAG into as few paths as possible. In the version with boxes, the vertex set is the set $\{1, \ldots, n\}$ of boxes, each edge (i, j) is present if and only if i fits inside j, and each path represents a valid nesting of boxes. Design a $O(|V|^3)$-time algorithm for this path partitioning problem.

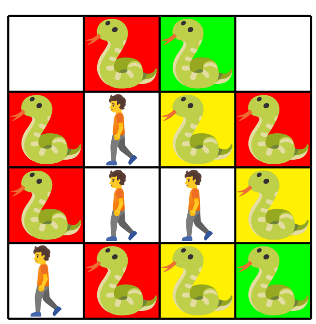
   Hint 1: Minimizing the number of paths in your partition is equivalent to maximizing the number of edges used by paths in your partition. That is, minimizing the number of trips is equivalent to maximizing the number of times we put a box directly inside another box.

   Hint 2: Include two vertices in your flow network for each vertex in the original DAG.

3. Consider an $n \times n$ board with k hikers and ℓ snakes, all (initially) on separate squares. Each snake is initially *asleep*, *awake*, or *angry*, and these states are known.

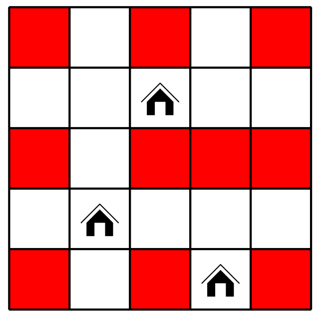
   The hikers can move vertically and horizontally. The snakes are stationary. If a hiker steps on the square of an asleep snake, the snake will wake up. If a hiker steps on the square of an awake snake, the snake will become angry. If a hiker steps on the square of an angry snake, the snake will bite the hiker. Each hiker's goal is to *escape*, which means reaching the bottom edge of the board without being bitten. Design a $O(kn^2)$-time algorithm that determines whether it is possible for all k hikers to escape.

   For example, if green means asleep, yellow means awake, and red means angry, then your algorithm should return FALSE for the board below.



4. A housing developer has purchased a large, square piece of land, divided up into an $n \times n$ grid of square sites. Only some of these sites are suitable for building houses. The ocean is to the east, and there are beautiful mountains to the north, so it is important that every house has unobstructed northward and eastward views. That is, no house can be directly north or directly east of any other house.

   The diagram below, where unsuitable sites are marked red, shows one valid arrangement of houses. With this same set of suitable sites, it is also possible to validly arrange four houses, but there is no valid arrangement of five houses.

   Design a $O(n^3)$-time algorithm that takes as input the coordinates of the suitable sites and returns the maximum number of houses that can be validly arranged.