**CS 401: Computer Algorithms I** **Fall 2017**

# Homework 2 (Due October 6)

# No Late Submissions

## 1. Graph Algorithms

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are starting to be used for high resolution and 3D mapping[1]. A group of ecologists have decided to use a (small) UAV to map out their study area, down to every tree. Unfortunately, it is hard to tell from the resulting image data the total number of species of trees in the area since many look similar.[2] Suppose there are $n$ trees in the area and let us assume for simplicity that there are only two different species, say Acacia and Bactris. Believe it or not, it is very hard to tell by looking at a tree by itself which kind it is. However, it is much easier to tell whether two trees are the same species or not. So the scientists do the following.

For each pair of 3D tree images $i$ and $j$ the scientists look at them side by side and decide whether they are the "same" species or "different". They also have the option of not giving an opinion and just leave the pair without a decision.

So now the scientists have the collection of $n$ trees, as well as a collection of $m$ decisions (either "same" or "different") for the pairs for which some decision was made. They would like to know whether this data is self-consistent. That is, we will say that $m$ decisions are *consistent* if it is possible to reliably label each tree either Acacia or Bactris in such a way that for each "same" pair $(i, j)$ the trees $i$ and $j$ indeed have the same label, while for each "different" pair $(i, j)$ the trees $i$ and $j$ have different labels.

*Give an algorithm with running time $O(m+n)$ that determines whether the $m$ decisions are consistent. Don't forget to prove its correctness and termination. Note that the input consists of the number of trees, $n$, and a list of $m$ decisions for some pairs of trees. For example, $n = 4$, and $(1, 2)$ same, $(1, 3)$ different, $(2, 4)$ same.*

**Extra Credit:** Will your algorithm work if there are more than two species, say three (with Cecropia)? If it does, give a proof of correctness. If not, give a counter-example.

## 2. Dijsktra

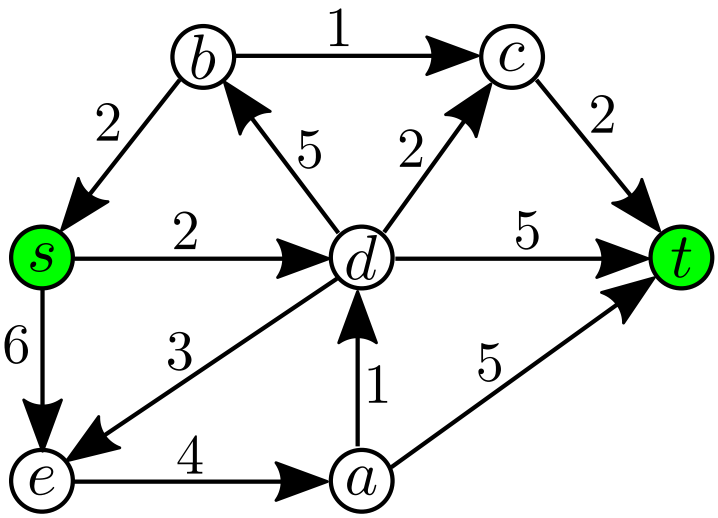
Show, step by step, how Dijkstra's algorithm works on the graph below, with a source $s$ and a destination $t$. What is the shortest path from $s$ to $t$?

## 3. Finding a cheap flight

Imagine you graduated and got a job at stingyflights.com. Your task is to design an algorithm for finding cheap flights. You are given a list of $n$ cities, and a list of $m$ flights

[1] E.g., `https://www.sensefly.com/applications/surveying.html`

[2] This is an existing ongoing project at UIC: `https://www.evl.uic.edu/entry.php?id=2228`



between them. For each flight $i$, you are given its departure city $x_i$, its destination city $y_i$, its departure time $t_i$, its duration $d_i$, and its cost $c_i$. So flight $i$ departs from city $x_i$ at time $t_i$, arrives at city $y_i$ at time $t_i + d_i$, and costs $c_i$ dollars.

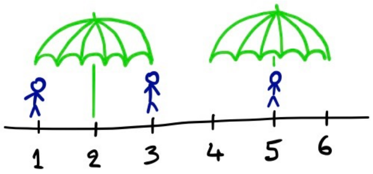
Design an algorithm that given a pair of cities $a$, $b$, computes the cheapest possible route (i.e. sequence of flights) that starts at $a$ at time 0, and ends at $b$ at time at most $T$. The running time of your algorithm should be polynomial in $n$ and $m$ (e.g. $O(n^{10}m^5)$).

Hint: Express the above problem as shortest-path computation in some graph.

## 4. A day at the beach

A group of $n$ people are lying on the beach. The beach is represented by the real line $\mathbb{R}$ and the location of the $i$-th person is some integer $x_i \in \mathbb{Z}$. Your task is to prevent people from getting sunburned by covering them with umbrellas. Each umbrella corresponds to a closed interval $I = [a, a + L]$ of length $L \in \mathbb{N}$, and the $i$-th person is covered by that umbrella if $x_i \in I$. Design a greedy algorithm for covering all people with the minimum number of umbrellas. The input consists of the integers $x_1, \ldots, x_n$, and $L$. The output of your algorithm should be the positions of umbrellas.

For example, if the input is $x_1 = 1$, $x_2 = 3$, $x_3 = 5$, and $L = 2$, then an optimum solution is the set of two umbrellas placed at positions 2 and 5, covering intervals $[1, 3]$ and $[4, 6]$.



The running time of your algorithm should be polynomial in $n$.