EECS 340 Algorithms

2018 Spring Semester

Homework 6

Due on April 25th, 12:45pm

Topics: Shortest paths, minimum spanning trees, and approximation algorithms.

1. **Warm-Up**

   (a) Solve R-15.1 in the text. $(\star)$

   (b) Solve R-18.3 in the text. $(\star)$

2. **Dijkstra Revisited**

   Solve C-14.3 in the text. $(\star\star)$

3. **The Struggle of Commuter Students**

   Given an undirected weighted graph $G = (V, E)$ with non-negative edge weights, and two vertices $s, t \in V$, let G' be G, but where every edge has its weight increased by 1. Prove that G and G' have the same set of shortest paths from s to t, or provide a counterexample. $(\star\star)$

4. **Unique Bonsai**

   Solve C-15.3 in the text. $(\star\star)$

5. **Bonsai No More**

   (a) Solve C-15.9 in the text. $(\star)$

   (b) In the setting of C-15.9, suppose that instead of adding 1 to the weight of every edge, we multiply the weight of every edge by 2. Are the minimum spanning trees of G and G' still the same? Prove, or provide a counterexample. $(\star)$

6. **Burma Shave**

   You've learned, much to your dismay, that the LED text display in the Glennan Student Area has been replaced. In its place is a new-fangled and expensive iLED display produced by Plum Inc. which has reduced functionality, only capable of inserting, deleting, or replacing one character per second [1]. It has a very nice touch-screen interface on the back, however, which lets you easily enter a list of strings for

[1] But dang, do those string transitions look pretty.

it to cycle through in sequence. You've been tasked with writing a program which takes as input a set of strings and returns an order to display the strings which minimizes the total transition time between strings in each cycle. A wise elder in HackCWRU informs you that it may not be tractable to achieve the minimum total transition time, but you should be able to come up with a somewhat bearable solution which is within a factor of two of the minimum.

(a) Describe an algorithm to do this which has polynomial runtime in the total length of the strings. Assume that you have access to a method `TimeBetween`(S_1, S_2) which computes the minimum number of seconds it would take to transition between the string S_1 and the string S_2 and runs in $O(|S_1| * |S_2|)$ time. $(\star \star \star)$
Hint: If we have three strings S_1, S_2, S_3, how do `TimeBetween`(S_1, S_2), `TimeBetween`(S_2, S_3), and `TimeBetween`(S_1, S_3) relate?

(b) **Extra Finals Practice (Not Graded):**
Implement `TimeBetween` using dynamic programming. As an example of `TimeBetween`'s expected behavior, if we let $S_1 =$"arrhythmic" and $S_2 =$"algorithmic" [2], then one sequence of edit operations we could perform to get from S_1 to S_2 would be:

String	Edit Operation
arrhythmic	
alrhythmic	(replace "r" with "l")
algrhythmic	(insert "g")
algorhythmic	(insert "o")
algorythmic	(delete "h")
algorithmic	(replace "y" with "i")

which turns out to have the minimal number of edit operations (5), so `TimeBetween` would return 5. $(\star \star \star)$

7. **FedUP, Regulated Federally**

   Solve A-18.4 in the text. $(\star\star)$

8. **The following is for EECS 454 students only**.

   Read Section 17.5 in the textbook and then solve the following problems:

   (a) R-17.3

   (b) R-17.14

Submission

Hand in your paper in-class by the beginning of class on the due date. Always check Canvas for updates and corrections.

[2] Our mixtape, "Algorhythmic" will be dropping this summer, brought to you by the Applicative Funktors.