

Recitation: Week 14

EE4033 Algorithms, Fall 2019

Instructor: Yao Wen Chang, James Chien Mo Li, and Iris Hui Ru Jiang

Presenter: Yi-Ting Lin, Yu-Sheng Lu



HW4

- Deadline: 12/17(Tue.) 9:00 AM
 - Submit your solution 10 minutes before the class begins
- Collaboration policy
 - Please specify all of your collaborators (name and student id) for each problem. If you solve some problems by yourself, please also specify “no collaborators”. **Problems without collaborator specification will not be graded.**

Problem 1 (Exercise 23.1-11)

23.1-11 *

Given a graph G and a minimum spanning tree T , suppose that we decrease the weight of one of the edges not in T . Give an algorithm for finding the minimum spanning tree in the modified graph.

- Give an algorithm. Briefly justify the correctness.
 - Do not directly find the MST in the modified graph.

Problem 2 (Exercise 23.2-7)

23.2-7 *

Suppose that a graph G has a minimum spanning tree already computed. How quickly can we update the minimum spanning tree if we add a new vertex and incident edges to G ?

- Give an algorithm. Briefly justify the correctness.
 - Do not directly find the MST in the modified graph.

- How quickly?
 - Original graph $G = (V, E)$
 - New graph $G' = (V \cup \{v\}, E \cup E_v)$

Problem 3 (Exercise 23.4)

23-4 Alternative minimum-spanning-tree algorithms

In this problem, we give pseudocode for three different algorithms. Each one takes a connected graph and a weight function as input and returns a set of edges T . For each algorithm, either prove that T is a minimum spanning tree or prove that T is not a minimum spanning tree. Also describe the most efficient implementation of each algorithm, whether or not it computes a minimum spanning tree.

a. MAYBE-MST-A(G, w)

```
1 sort the edges into nonincreasing order of edge weights  $w$ 
2  $T = E$ 
3 for each edge  $e$ , taken in nonincreasing order by weight
4   if  $T - \{e\}$  is a connected graph
5      $T = T - \{e\}$ 
6 return  $T$ 
```

b. MAYBE-MST-B(G, w)

```
1  $T = \emptyset$ 
2 for each edge  $e$ , taken in arbitrary order
3   if  $T \cup \{e\}$  has no cycles
4      $T = T \cup \{e\}$ 
5 return  $T$ 
```

c. MAYBE-MST-C(G, w)

```
1  $T = \emptyset$ 
2 for each edge  $e$ , taken in arbitrary order
3    $T = T \cup \{e\}$ 
4   if  $T$  has a cycle  $c$ 
5     let  $e'$  be a maximum-weight edge on  $c$ 
6      $T = T - \{e'\}$ 
7 return  $T$ 
```

- For each algorithm,
 - Prove T is a MST or not
 - Describe the most efficient implementation

Problem 4 (Exercise 24.1-6)

24.1-6 *

Suppose that a weighted, directed graph $G = (V, E)$ has a negative-weight cycle. Give an efficient algorithm to list the vertices of one such cycle. Prove that your algorithm is correct.

- Give an **efficient** algorithm.
- Prove that your algorithm is correct.

Problem 5 (Exercise 24.2-4)

24.2-4

Give an efficient algorithm to count the total number of paths in a directed acyclic graph. Analyze your algorithm.

- Give an **efficient** algorithm. Briefly justify the correctness.
- Analyze your algorithm.

Problem 6 (Exercise 24.4-1)

24.4-1

Find a feasible solution or determine that no feasible solution exists for the following system of difference constraints:

$$x_1 - x_2 \leq 1,$$

$$x_1 - x_4 \leq -4,$$

$$x_2 - x_3 \leq 2,$$

$$x_2 - x_5 \leq 7,$$

$$x_2 - x_6 \leq 5,$$

$$x_3 - x_6 \leq 10,$$

$$x_4 - x_2 \leq 2,$$

$$x_5 - x_1 \leq -1,$$

$$x_5 - x_4 \leq 3,$$

$$x_6 - x_3 \leq -8.$$

- Please show your steps.

Problem 7 (Problem 24-3)

24-3 Arbitrage

Arbitrage is the use of discrepancies in currency exchange rates to transform one unit of a currency into more than one unit of the same currency. For example, suppose that 1 U.S. dollar buys 49 Indian rupees, 1 Indian rupee buys 2 Japanese yen, and 1 Japanese yen buys 0.0107 U.S. dollars. Then, by converting currencies, a trader can start with 1 U.S. dollar and buy $49 \times 2 \times 0.0107 = 1.0486$ U.S. dollars, thus turning a profit of 4.86 percent.

Suppose that we are given n currencies c_1, c_2, \dots, c_n and an $n \times n$ table R of exchange rates, such that one unit of currency c_i buys $R[i, j]$ units of currency c_j .

Problem 7 (Problem 24-3) conti.

- a. Give an efficient algorithm to determine whether or not there exists a sequence of currencies $\langle c_{i_1}, c_{i_2}, \dots, c_{i_k} \rangle$ such that

$$R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1] > 1 .$$

Analyze the running time of your algorithm.

- b. Give an efficient algorithm to print out such a sequence if one exists. Analyze the running time of your algorithm.

- Give an **efficient** algorithm. Briefly justify the correctness.
- Analyze your algorithm.

Problem 8 (Exercise 25.2-1)

25.2-1

Run the Floyd-Warshall algorithm on the weighted, directed graph of Figure 25.2. Show the matrix $D^{(k)}$ that results for each iteration of the outer loop.

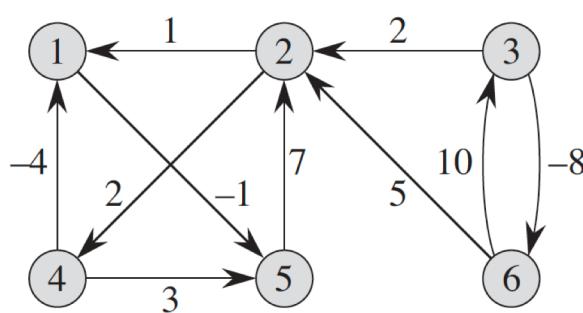


Figure 25.2 A weighted, directed graph for use in Exercises 25.1-1, 25.2-1, and 25.3-1.

- Please show your steps. (each matrix $D^{(k)}$ for all k)

Problem 9 (Exercise 25.3-4)

25.3-4

Professor Greenstreet claims that there is a simpler way to reweight edges than the method used in Johnson's algorithm. Letting $w^* = \min_{(u,v) \in E} \{w(u, v)\}$, just define $\hat{w}(u, v) = w(u, v) - w^*$ for all edges $(u, v) \in E$. What is wrong with the professor's method of reweighting?

- What is wrong?
 - Demonstrate on a counterexample