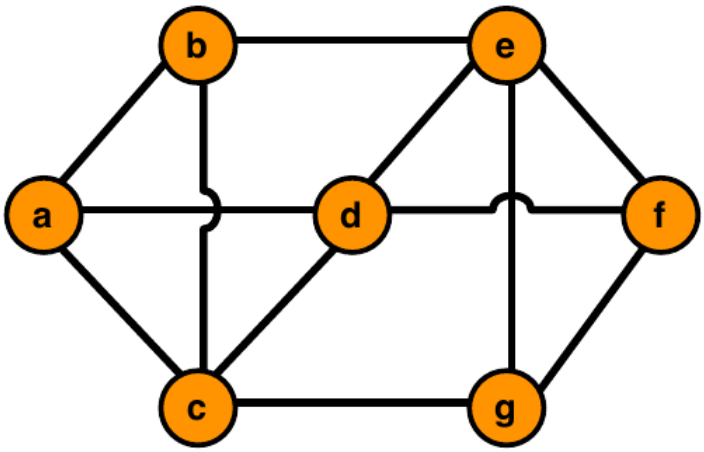
CSCI 3104-300 Summer 2019 Problem Set 7
Hoffman, Ryan
01/11

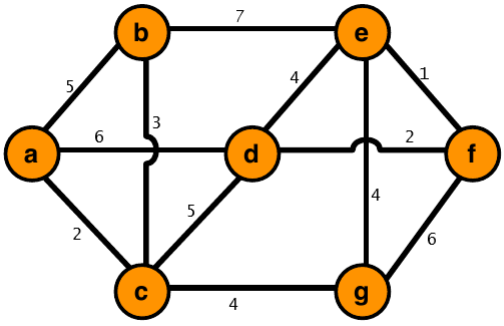
1. (30 points) Grog gives you the following unweighted graph and asks you to construct a weight function w on the edges, using positive integer weights only, such that the following conditions are true regarding minimum spanning trees (MST) and single-source shortest path trees (SSSP):

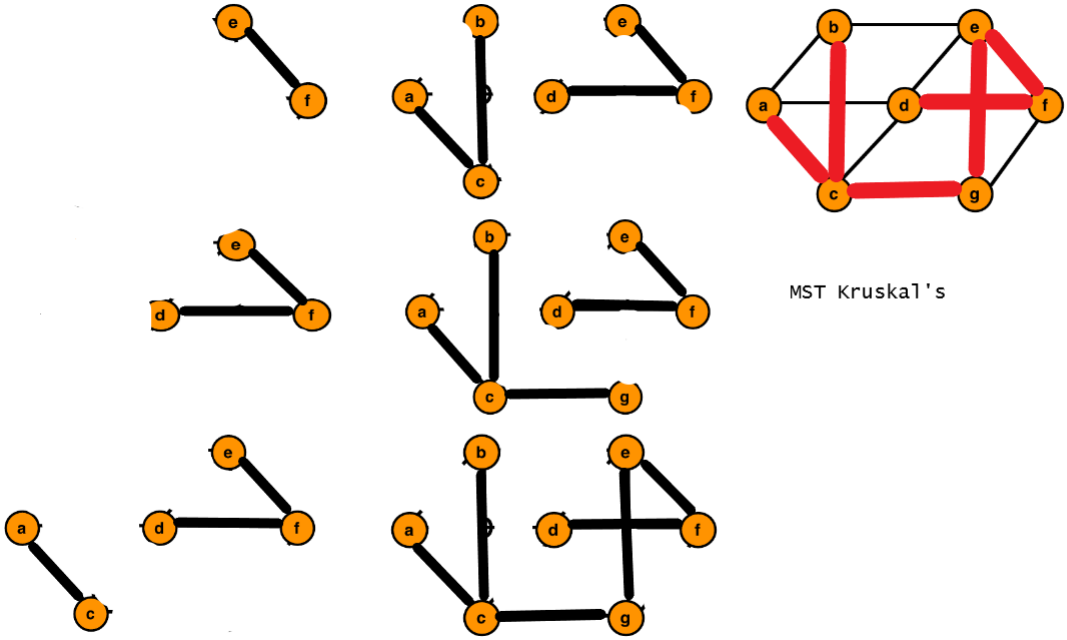
   - The MST is distinct from any of the seven SSSP trees.
   - The order in which Jarník/Prim's algorithm adds the safe edges is different from the order in which Kruskal's algorithm adds them.

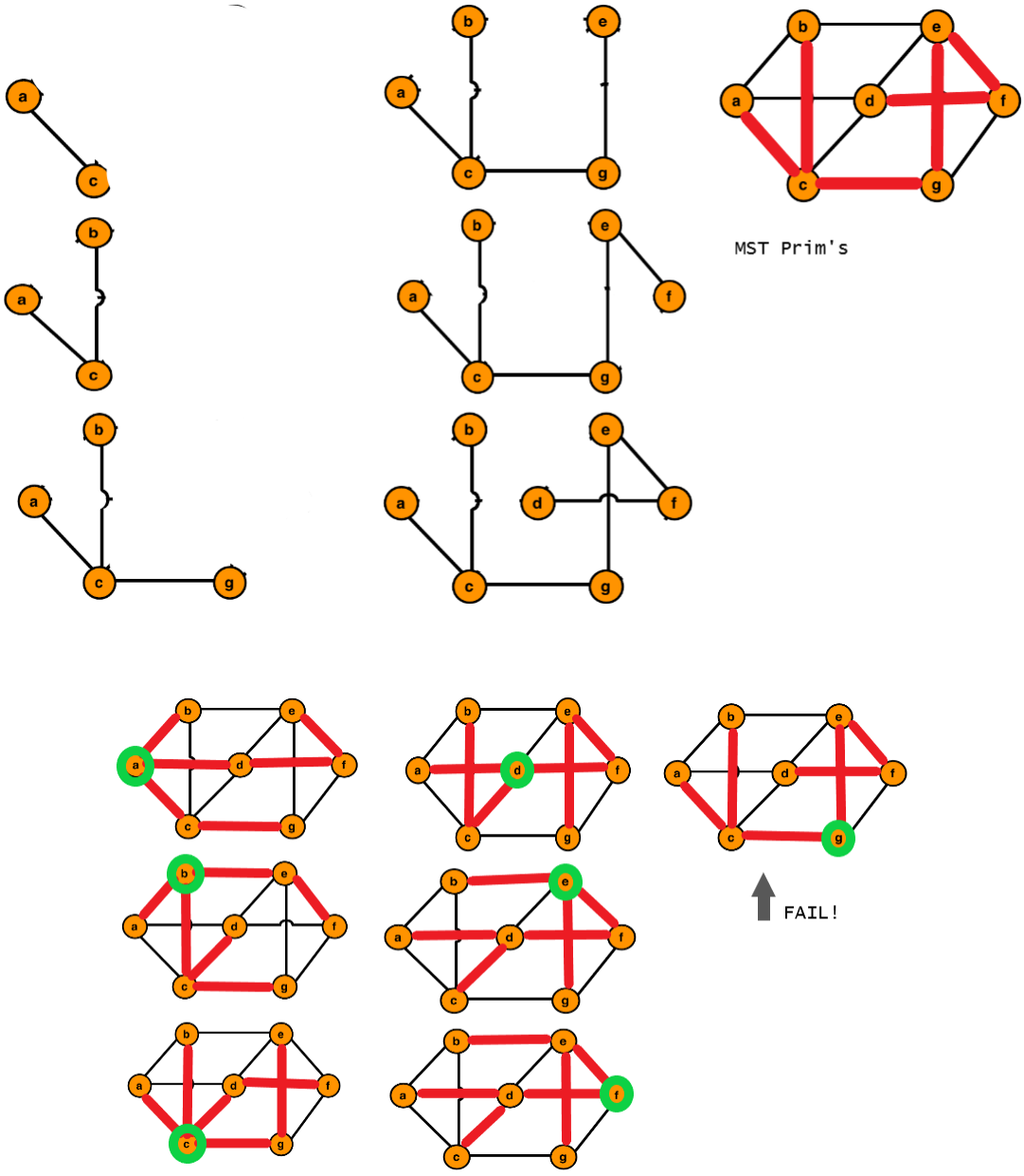
   Justify your solution by (i) giving the edges weights, (ii) showing the corresponding MST and all the SSSP trees, and (iii) giving the order in which edges are added by each of the two algorithms.



Solution:







2. (25 points) Harry and Shadow think they have come up with a way to get rich by exploiting the ore market.

   (a) When given R and α, give an efficient algorithm that can determine if an arbitrage

opportunity exists. Analyze the running time of your algorithm. Thormund's hint: It is possible to solve this problem in $O(n^3)$.

Solution:

Using Bellman-Ford to determine if a negative-weight cycle exists. To do this, however, we need to introduce a vertex s, with weight 0 from s to all vertices. This way, we can detect a negative cycle if it exists. We also need a weight function that takes into account the given transaction cost α.

Here is the weight function:

$$w(v_i, v_j) = \lg \frac{1}{R[i,j]\alpha}$$

$$= -\lg R[i,j]\alpha$$

This gives us a directed negative-weighted graph to work with[1].

Now, since Bellman-Ford returns *True* if no negative-weight cycle exists and *False* otherwise, we just need to invert the result.

Analysis:

To create the directed, negative-weighted graph is $\Theta(n^2)$

Running Bellman-Ford in the above algorithm is $O(n^3)$.[2]

(b) For an arbitrary R, explain how varying α changes the set of arbitrage opportunities that exist and that your algorithm might identify.

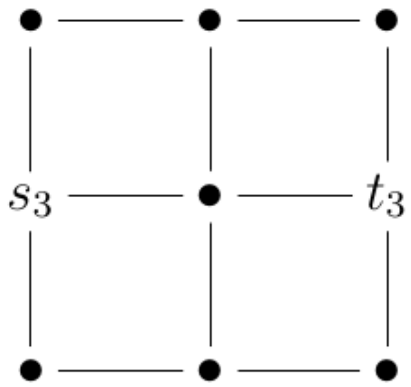
Solution:

Variations on the transaction cost, α, affects the no-arbitrage condition by creating an area (or region) where arbitrage is not profitable.[3]

3. (a) **Solution:**
Not able to produce a solution.

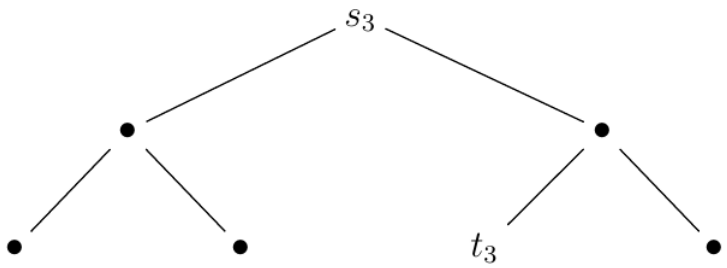
(b) **Solution:**
Not able to produce a solution.

(c) i. Grids.



Solution:
Not able to produce a solution.

ii. Trees.



Solution:
Not able to produce a solution.

iii. Random graphs.
Solution:
Not able to produce a solution.

References

[1] https://www.dailycodingproblem.com/blog/how-to-find-arbitrage-opportunities-in-python/

[2] http://www.cs.princeton.edu/courses/archive/spring06/cos226/lectures/shortest-path.pdf

[3] https://pdfs.semanticscholar.org/7285/359f5cf6cac25a54c4b2126a090352bf1fe4.pdf