/

U C L A Computer Science Department

CS 180 **Algorithms & Complexity** **ID (4 digit):**

Midterm **Total Time: 1.5 hours** **February 2016**

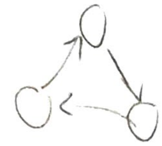
(19)

Each problem has 25 points.

1 **A.** Describe Topological Sort (in English, bullet by bullet) on a DAG **B.** Analyze its time complexity and justify your answer. **C.** What will happen when there is a cycle? Prove your answer.

A) • while there are still unvisited nodes in the graph G:

- remove an arbitrary source node N and its edges from the graph ✓
- order the source node N first in the list
- run the steps above again for G minus N and its edges, and place the results after N in the ordered list



B) $O(|V|+|E|)$ $O(V^2)$ (-5)

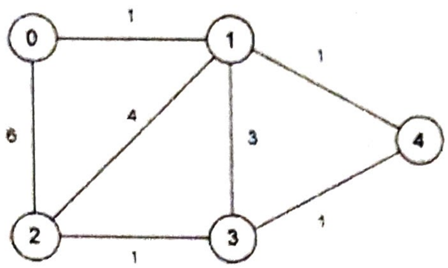
There are |V| nodes in the graph that must be visited and sorted, and each iteration through the loop must check a portion of the |E| edges to find a source node.

(-1) initial step

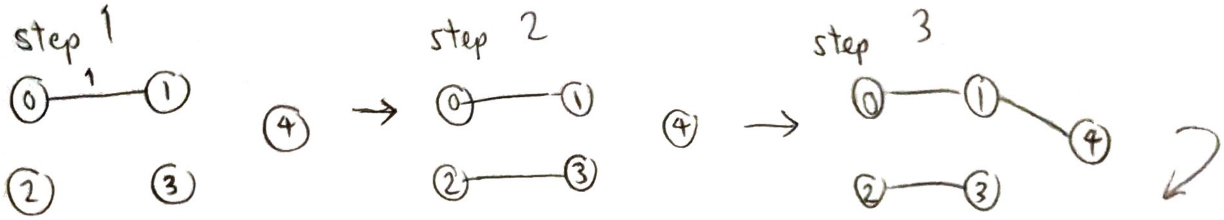
C) If the sort algorithm reaches a cycle, there will be no source node in that subgraph of G since every node must point to one another in some way. The sort will fail to find a source node and will get stuck with an incomplete sorted DAG. ✓

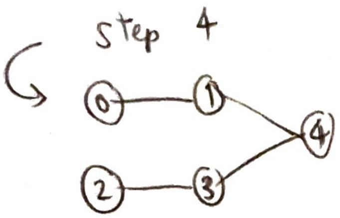
2 1

2 . A. Use Kruskal's MST algorithm to find a an MST in this graph. Show each step.
B. If some edges were negative would the algorithm still find an MST. Prove your answer.



A)





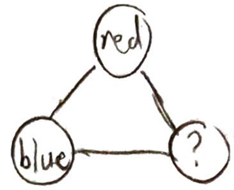
check for cycle? sort edges,

B) Yes, since Kruskal's is a greedy algorithm. Given a graph G with a negative cost edge i and positive cost edges j, k, where i, j, k form a 3-cycle, [illegible] assume the algorithm picks j and k for the MST. However, that means that the algorithm has picked the lowest cost edges that would not form a cycle at each step, so i could not be negative. This would contradict the given graph, so the algorithm must choose i first. why is kruskal immune to < 0 cost?

19

3. A graph is two-colorable if we can color its vertices with RED & BLUE such that no two adjacent vertices have the same color. **A.** Are all graphs two-colorable. Prove your answer **B.** Design an efficient algorithm for two coloring a graph. Prove the correctness of your algorithm. **C.** Analyze its time complexity. **D.** How many colors do you need to color a tree?

A) No, counterexample:



← will have 2 same-color adjacent vertices no matter which color is chosen

B)

- have a variable COLOR that can be flipped between red and blue
- keep a set of visited nodes in the graph
- starting at an arbitrary node N, while there are still unvisited nodes:
  - at each layer 0, 1, 2, etc. nodes away from N:
  - set each unvisited node in the layer to COLOR and mark it as visited
  - when finished w/ the layer, flip COLOR and move to layer +1

-1 check 2-colorable

proof? -5

C) $O(|V| + |E|)$

The algorithm uses a breadth first search that follows each edge away from a node up to $|E|$, ~~to~~ all nodes $|V|$ to color every single one.

D) 2 colors - A tree has no 3-cycles (or any cycles at all) that require an extra color for the 3rd node. A child node can always be the opposite color of the parent.

4. Consider a sequence x1, x2, ... , xn of (positive and negative) integers. We want to find two indices i and j such that xi + ... + xj is maximized.

A. Describe in English (bullet by bullet) an O(n) time algorithm (using constant extra space) for solving this problem. For example, if the input is (-2, -2, 5, 7, -3, 4, -4) then i=3 and j = 6 (and the sum xi+…+xj is equal to 13). **B**. Prove the correctness of your algorithm.

4/25

A)
- have 2 indices i and j initially set to 0
- have a variable MAX tracking the maximum sum found, and a SUM tracking the sum so far
- for each x in the sequence x_1 to x_n:
  - add x to SUM
  - if SUM > MAX:
    - set j to the index of x
    - set MAX to SUM
- for each x in the sequence x_j to x_1 (going backward):
  - subtract x from SUM
  - if SUM > MAX:
    - set i to the index of x
    - set MAX to SUM

B) The algorithm should find a subsequence with the maximum sum, as no numbers can be omitted between i and j. This means that only one or two passes through the list, narrowing down from the left and right of the sequence, is needed to find the maximum sum. If an x is negative and removed from the seq, then the sum increases.