CS:5620 Midterm Exam, Fall 2016

Tuesday, Oct 11, 2016

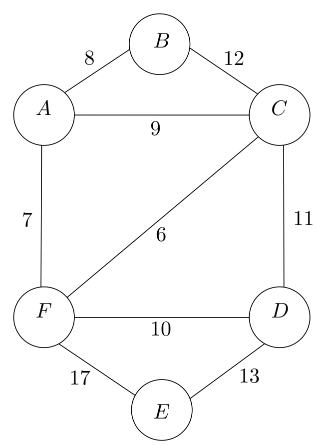
---

1. (a) You are given a 6-node cycle with nodes $v_1, v_2, \ldots, v_6$. For $1 < i < 6$, node v_i is connected to nodes v_{i-1} and v_{i+1}. Node v_1 is connected to nodes v_6 and v_2 and node v_6 is connected to nodes v_5 and v_1. The nodes have 16-bit IDs as shown below:

v_1	0111 1011 1101 1110
v_2	0110 1011 1101 1110
v_3	0111 1010 1111 1110
v_4	0111 1011 1101 1100
v_5	0111 1010 1100 1110
v_6	0100 1011 1101 1110

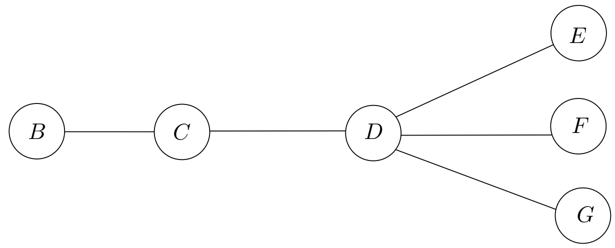
Recall that in the Cole-Vishkin algorithm for graph coloring, node IDs are taken to be initial colors. Write down the colors of nodes v_1, v_2, and v_3 after the excution of one round of the Cole-Vishkin algorithm.

(b) Consider the execution of the Distributed Kruskal algorithm on the following graph. Suppose that the BFS tree we use is rooted at B and the rest of the BFS tree is defined by $parent(A) = B$, $parent(C) = B$, $parent(F) = C$, $parent(D) = C$, and $parent(E) = D$. Write down the edge sent by each non-root node to its parent during the first round of execution of the Distributed Kruskal algorithm. Then, write down the edges discarded from the set E_v by each node v, also in the first round.



2. (a) You are given an n-node cycle. Fix an arbitrary node v. Calculate the probability that v becomes permanently colored in the first round of execution of Luby's $(\Delta + 1)$-coloring algorithm. What is the expected number of nodes that will participate in the second round of the algorithm?

(b) Suppose that Luby's MIS algorithm is executed on the following graph. Calculate the exact probability that node D becomes inactive after the execution of the first round of Luby's algorithm. Recall that a node becomes inactive if it joins the MIS or if one or more of its neighbors joins the MIS. Also recall that in the version of Luby's algorithm discussed in class, if two neighbors decide to tentatively join the MIS, the one with higher degree wins; and if degrees are identical then IDs are used to break symmetry.



3. (a) Using Linial's lower bound proof, derive a lower bound on the round complexity of any deterministic algorithm in the LOCAL model that colors an oriented cycle using 32 colors (instead of 3).

   Note: (a) If you are allowed more colors, it should become easier to color a graph and so you should expect a weaker (smaller) lower bound. (b) To get a lower bound for 32-coloring an oriented cycle, you just need to modify a small part of Linial's lower bound proof. This is all you should write in your answer.

   (b) Just before some round in Luby's algorithm for distributed $(\Delta + 1)$-coloring, the state of the neighborhood of some uncolored node u is as follows:

   (i) u has 3 uncolored neighbors v, w, x.
   (ii) $P(u) = \{1, 2, 3, 4\}$.
   (iii) $P(v) = \{1, 3, 4\}$, $P(w) = \{1, 4\}$, and $P(x) = \{1, 3\}$.

   In the proof of Proposition 5 in Johansson's analysis of this algorithm, it is claimed that if we pick a color c from $P(v)$ that minimizes $Pr(\overline{W}_{c,N(u)\setminus\{v\}})$ and remove this color from $P(v)$, this will not increase $Pr(C_u)$. Identify a color in $P(v)$ to remove, using this rule. Calculate $Pr(C_u)$ before the color-removal and after the color-removal.

4. You have a distributed system with 10 machines that can directly communicate with each other. Thus the underlying communication network is a size-10 clique (complete graph). You may assume that the machines have distinct IDs: 1 through 10. You want to use this system to compute an MST of an edge-weighted graph G with 100,000 nodes and 10 million edges. The 10 million edges of G are distributed evenly among the 10 machines and so each machine is given one million edges (along with their weights). Besides the fact that the distribution of edges is even, you cannot make any other assumptions about the edge distribution (for example, you cannot assume that the lightest million edges go to machine 1!).

   Assume that the 100,000 nodes of G are uniquely numbered 1 through 100,000. Thus each edge $\{u, v\}$ can be represented by a 3-tuple:

$$(u\text{'s number}, v\text{'s number}, \text{weight of edge } \{u, v\}).$$

   Assume that edge-weights are not too large and the 3-tuple representing an edge can be stored in a single *word* of memory (e.g., a 64-bit word).

   In each round, a machine can send one message to any one of the other 9 machines. However, this message can be fairly large – of size at most 10,000 words. Similarly, in each round a machine can receive a 10,000-word message from any one of the other 9 machines. In other words, a machine cannot, in one round, receive messages from or send messages to two or more distinct machines. For example, in some round it may be the case that machine 1 sends a message to machine 2, machine 2 sends a message to machine 3, machine 3 sends a message to machine 4, and so on and machine 10 can send a message to machne 1. Note that in this example, every machine sends a message to exactly one machine and every machine receives a message from exactly one machine.

   It is required that when the algorithm ends, the MST (i.e., all edges of the MST) are known to machine 1. Your task is to design an algorithm that solves this problem in 40 rounds.

   Note: Even though the model of computation in this problem is neither the LOCAL model nor the CONGEST model, we can still use ideas from CONGEST model algorithms to solve this problem. Specifically, ideas used in Distributed Kruskal's algorithm are useful here.