CS31005 Algorithms – II, Autumn 2021–2022

First Test

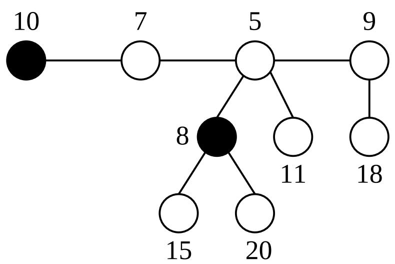
Sep 07, 2021, 2:15pm — Maximum marks: 50

Duration: 70 minutes (55 minutes for answering questions + 15 minutes for download/submission)

All your answers MUST BE HANDWRITTEN on paper. Scan all papers with your answers in a SINGLE pdf, and upload it as the answer to the Quiz in Moodle. The size of the final pdf must be less than 10 MB. You must upload the pdf strictly by 3:25 pm Moodle server time, the quiz submission will close after that.

---

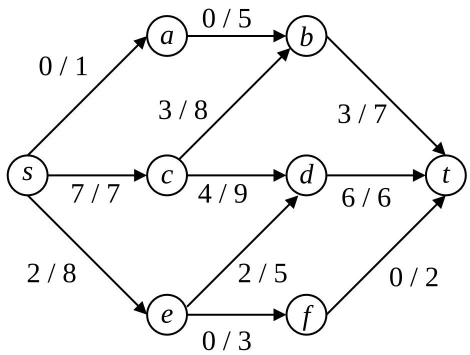
1. Consider the following Fibonacci Heap (the values stored at each node are shown against each node, and the marked nodes are shown as black). Show the new heap after ExtractMin is called on the heap (just show the new heap, no explanation is needed). Whenever you have a choice of going left or right in the root list, go right. **(5)**



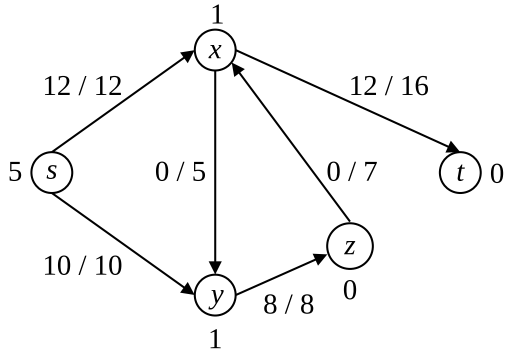
2. Consider the following flow network, with a feasible flow as shown. Here, s is the source, t is the sink, and x/y against a directed edge (u,v) shows $f(u,v)/c(u,v)$. Show

(i) the residual graph,
(ii) an augmenting path (pick the augmenting path with the highest residual capacity), and
(iii) the new flow network after the augmentation.

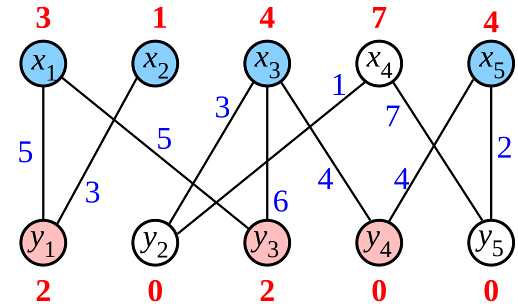
Show the augmenting path as a sequence of vertices, do not mark on the graph. For the flow network, show only the flows on actual edges of the graph, no need to show f values for vertex pairs that are not edges in the graph. No explanation is needed. **(2 + 2 + 2)**



3. Consider the following intermediate state of a flow network while executing the preflow-push method. Here, s is the source, and t is the sink. The height function values for each node are shown beside the node, and x/y against each edge indicates that the edge has a capacity y and a flow value x. However, the generic preflow-push method is modified such that (i) a push operation will be done at a step only if no relabel can be done on any node at that step, and (ii) if push operation is enabled at multiple nodes at a step, the push operation at the node with the highest excess will be done first. First, list the operations that will be done IN ORDER on the nodes (just list as push(i,j), relabel(i) etc. for nodes i,j, no need to show any details) until two push operations are done. Then, draw the final network showing the flow values on the edges and the height values at nodes after two push operations are done. No explanation is needed. **(6)**



4. Consider the weighted bipartite graph below (dummy edges with weight 0 are not shown, but they are there). The picture shows the vertex labels (shown in red against each vertex). Suppose that with respect to some partially computed matching in the equality graph, we have a need to relabel the vertices when $S = \{x_1, x_2, x_3, x_5\}$ and $T = \{y_1, y_3, y_4\}$. What will be the relabeled values (just show the values for each node, no explanation is needed)? Also show the equality graph *after* the relabeling. **(4 + 4)**



5. Consider a $p \times q$ matrix X. Each element of X is a non-negative real number. However, the sum of the elements in any row and in any column is an integer (may be different for different rows and columns). Your task is to construct a $p \times q$ matrix Y such that each element of Y is a non-negative integer, and the sum of the elements in each row or in each column in Y is the same as the sum in the corresponding row or column (respectively) in X. You must use maximum flow concepts to construct Y. **(10)**

6. Consider a large task T that can be decomposed into n subtasks $T_1, T_2, \ldots, T_n$. The subtasks are to be performed on two machines M_1 and M_2. A subtask can be done on any of the two machines, but will incur different costs on the two machines. Specifically, subtask T_i incurs a cost C_i if it is done on machine M_1, and cost D_i if it is done on machine M_2. Also, some pairs of these subtasks need to communicate among themselves. If such a pair of subtasks T_i and T_j are run on the same machine, the communication cost incurred is 0, otherwise, they incur a communication cost of P_{ij}. If a pair of subtasks do not need to communicate, they incur no communication cost irrespective of which machines they are run on. Find an assignment of the subtasks to the two machines so that the total cost of executing T is minimized. (**Hint:** Reduce the problem of finding the minimum cost of executing T to that of finding a minimum cut in a graph that you construct. First show the construction of the graph clearly, stating what the nodes and the edges are, and then argue why the minimum cut in this graph will give the minimum cost.) **(15)**