Graph Algorithm
Autumn 2023, Midterm Exam
November 1, 5:00 – 8:00 pm

1. (20%) Suppose we perform a sequence of n operations on a data structure in which the ith operation costs i if i is an exact power of 4, and 1 otherwise. Determine the amortized cost per operation by using the following three methods, respectively.

   (a) Aggregate analysis.

   (b) Accounting method.

   (c) Potential method.

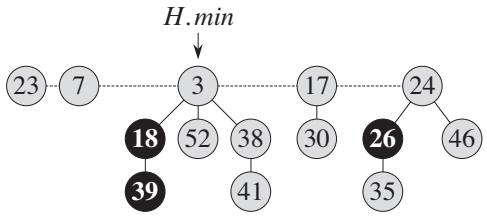
2. (20%) A Fibonacci heap is shown in the following figure. Problems $(a), (b), (c)$, and (d) are unrelated.

   (a) Show the results when a node with key 2 is inserted.

   (b) Show the results from calling Extract-Min.

   (c) Show the results when the node with key 39 has its key decreased to 2.

   (d) Show the results when the nodes with keys 39 and 21 have deleted in order.



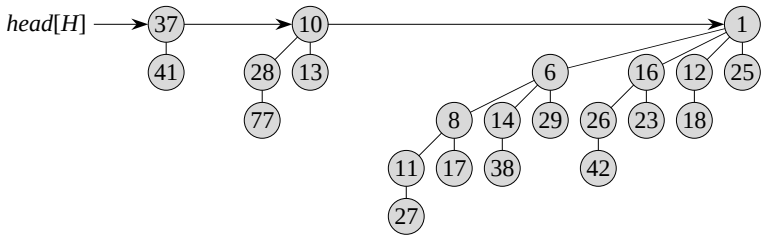
3. (20%) A binomial heap is shown in the following figure. Problems (c) and (d) are unrelated.

   (a) Show that the height of B_k is k .

   (b) Show that B_k has exactly $\binom{k}{i}$ nodes at depth i for $i = 0, 1, \ldots, k$.

   (c) Show the results when the node with key 8 has its key decreased to 2.

   (d) Show the binomial heap that results from calling Binomial-Heap-Extract-Min on the binomial heap.



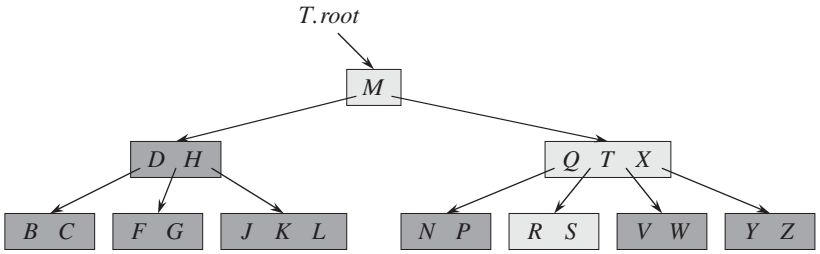
4. (20%) B-trees are balanced search trees designed to work well on magnetic disks or other direct-access secondary storage devices. Problems (c) and (d) are unrelated.

   (a) What is the maximum number of nodes in a B-tree of height 3 when t is 4?

   (b) What is the minimum number of nodes in a B-tree of height 3 when t is 4?

   (c) Show the results of inserting the keys U, A, E, and I in order into the following B-tree with minimum degree 2.

   (d) Show the results of deleting B and C, in order from the following B-tree with minimum degree 3.

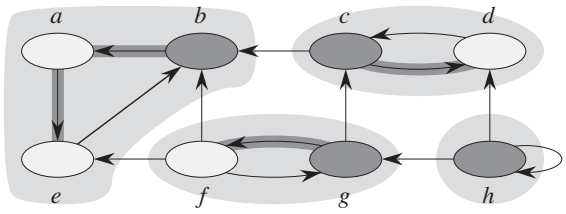


5. (20%) Show how depth-first search works on the graph of the following figure. Assume that the DFS procedure considers the vertices in reverse alphabetical order, and assume that each adjacency list is ordered in reverse alphabetical order.

   (a) Show the discovery and finishing times for each vertex.

   (b) Show the parenthesis structure of the depth-first search.

   (c) Show the classification of each edge.



6. (15%) Design an efficient algorithm (in terms of asymptotic complexity in the worst case) to determine if two students in a class of n students have the same height. What is the complexity of your algorithm? Consider the following two cases:

   (a) The smallest measure is centimeter.

   (b) There is no such a limit (i.e., each measure is precise).