

GRAPH ALGORITHM  
Autumn 2022, Midterm Exam  
November 3, 9:20 – 12:05 am

1. (20%) Suppose we perform a sequence of  $n$  operations on a data structure in which the  $i$ th operation costs  $i$  if  $i$  is an exact power of 3, 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. (15%) Suppose that CONNECTED-COMPONENTS is run on the undirected graph  $G = (V, E)$ , where  $V = \{a, b, c, d, e, f, g, h, i, j, k\}$  and the edges of  $E$  are processed in the order  $(d, i), (f, k), (b, g), (i, j), (g, i), (a, h), (d, k), (d, f), (g, d)$ . List the vertices in each connected component after each iteration of lines 3–5.

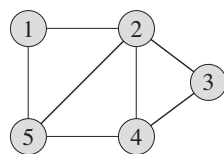
CONNECTED-COMPONENTS( $G$ )

```

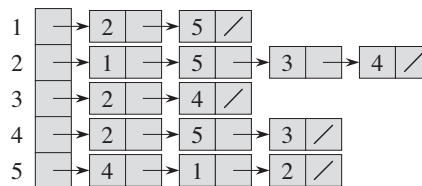
1  for each vertex  $v \in V$ 
2      MAKE-SET( $v$ )
3  for each edge  $(u, v) \in E$ 
4      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
5          UNION( $u, v$ )
  
```

| Edge processed | Collection of disjoint sets |         |         |            |         |            |         |         |         |         |         |
|----------------|-----------------------------|---------|---------|------------|---------|------------|---------|---------|---------|---------|---------|
| initial sets   | { $a$ }                     | { $b$ } | { $c$ } | { $d$ }    | { $e$ } | { $f$ }    | { $g$ } | { $h$ } | { $i$ } | { $j$ } | { $k$ } |
| $(d, i)$       | { $a$ }                     | { $b$ } | { $c$ } | { $d, i$ } | { $e$ } | { $f$ }    | { $g$ } | { $h$ } |         | { $j$ } | { $k$ } |
| $(f, k)$       | { $a$ }                     | { $b$ } | { $c$ } | { $d, i$ } | { $e$ } | { $f, k$ } | { $g$ } | { $h$ } |         | { $j$ } |         |

3. (15%) Illustrate the progresses of DFS, starting from vertex 2, on the following graph. Show the discovery and finishing times for each vertex, show the classification of each edge, and show the parenthesis structure.

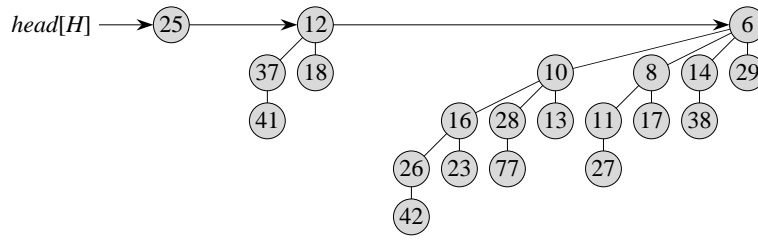


(a)

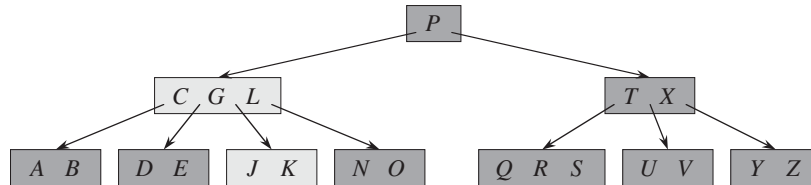


(b)

4. (15%) A binomial heap is shown in the following figure. Problems (a), (b), (c), and (d) are unrelated.



- (a) Show the results when a node with key 5 is inserted.
- (b) Show the results from calling EXTRACT-MIN.
- (c) Show the results when the node with key 42 has its key decreased to 2.
- (d) Show the results when the nodes with keys 12 has deleted.
5. (15%) A B-tree is shown in the following figure. Problems (a) and (b) are unrelated.
- (a) Show the results of inserting the keys  $F$ ,  $I$ , and  $H$  in order into the following B-tree with minimum degree 3.
- (b) Show the results of deleting  $G$ ,  $X$ , and  $Z$ , in order from the following B-tree with minimum degree 3.



6. (15%) Suppose you are given a sorted array of  $n$  distinct numbers that has been rotated  $k$  steps, for some unknown integer  $k$  between 1 and  $n - 1$ . That is, you are given an array  $A[1..n]$  such that the prefix  $A[1..k]$  is sorted in increasing order, the suffix  $A[k + 1..n]$  is sorted in increasing order, and  $A[n] < A[1]$ .
- For example, you might be given the following 16-element array (where  $k = 10$ ):

$$A = \langle 9, 13, 16, 18, 19, 23, 28, 31, 37, 42, -4, 0, 2, 5, 7, 8 \rangle.$$

Describe and analyze an algorithm to compute the unknown integer  $k$  in  $o(n)$  time. Yes, that is a little-oh.

7. (15%) The elements of an  $n$ -by- $n$  array  $A$  are arranged in ascending rows and columns; formally,  $A[i][j] < A[i][j']$  whenever  $j < j'$  and  $A[i][j] < A[i'][j]$  whenever  $i < i'$ . Suppose you wish to determine if  $A$  contains a particular target element  $x$ . Give the most efficient algorithm you can for solving this problem, and compute the best upper bound you can on its asymptotic worst-case running time as a function of  $n$ .