CS 2420 Algorithms and Data Structures

Fall Semester, 2021

Final Exam

3:00 p.m. – 4:50 p.m., Thursday, December 16, 2021

Your name and A-number:

Benjamin Swenson A02299890

Note:

1. **This exam has eight pages. There are three parts. The first two parts are both multiple choices. The third part is short answers.**

Total points: 100

Part I. Multiple Choices (2 points each, 30 points in total). Pick the single best answer.

1. A hash table is

   (a) a linked list $\boxed{\text{(b) an array}}$ (c) a binary tree (d) a queue (e) a stack

2. Two keys that originally hash onto different values compete for the same successive locations when resolving collisions. What collision resolution method has this **primary clustering** feature?

   (a) double hashing (b) quadratic probing (c) separate chaining $\boxed{\text{(d) linear probing}}$
   (e) a, b, and d (f) none of the above

3. In the following approaches to resolving collisions in hash tables, which approach needs other extra space than the hash table itself?

   (a) linear probing (b) quadratic probing (c) double hashing $\boxed{\text{(d) separate chaining}}$

4. In a min-heap, searching a key can be done in O(log n) time. Is this statement correct or not?

   (a) Yes $\boxed{\text{(b) No}}$

5. What is the time complexity for building a heap from an arbitrary array of n elements?

(a) O(log n) (b) O(n) (c) O(n log n) (d) $O(n^2)$

6. Given a min-heap, what is the time for finding the smallest key in the heap (without removing it from the heap)?

(a) O(1) (b) O(log n) (c) O(n) (d) O(n log n)

7. Given a min-heap, what is the time for removing the smallest key in the heap?

(a) O(1) (b) O(log n) (c) O(n) (d) O(n log n)

8. We want to use Bubble Sort to sort the following numbers 5 3 8 9 1 7 0 in ascending order. The algorithm will stop after at most n-1 rounds of swap operations. What is the array after the first round of swap operations?

3 5 8 1 7 0 9

(a) 5 3 8 1 7 0 9 (b) 3 5 8 1 0 7 9
(c) 3 5 8 9 1 7 0 (d) 3 5 8 1 7 0 9 (e) none of the above

9. Which of the following sorting algorithms does NOT require $O(n^2)$ time in the worst case?

(a) insertion sort (b) selection sort (c) heap sort (d) bubble sort (e) quick sort

10. Given a graph of n vertices and m edges, if we use adjacency lists to represent the graph, what is the space of the adjacency lists?

(a) O(n) (b) O(m) (c) O(m+n) (d) $O(n^2)$ (e) O(mn)

11. The breadth first search (BFS) on a graph is normally done using

(a) a stack (b) a first-in-first-out queue (c) a heap (d) a topological ordering
(e) none of the above

12. If we apply the DFS (depth-first-search) algorithm on a binary tree, starting from the root, what traversal is it with respect to the tree?

(a) pre-order (b) in-order (c) post-order (d) none of the above

13. A topological ordering of a graph G exists if

(a) G is a DAG (directed acyclic graph) (b) G is connected
(c) G is undirected (d) none of the above

14. Can Dijkstra's algorithm be used to compute shortest paths in graphs with negative edge weights?

(a) Yes (b) No

15. What is the maximum height of a tree in the union-find data structure using the strategy "union-by-height"?

(a) O(1) (b) O(log n) (c) O(n) (d) $O(n^2)$

Part II. Multiple Choices (3 points each, 30 points in total). Pick the single best answer.

1. Consider a hash table with linear probing and a size of 9. Use the hash function hash(x) = x % 9. Insert the keys: 7, 24, 6, 15, 0, 12 into your table (in that order). What is the result?

(a)

0	15		12		24	6	7	

(b)

0	15		12			24	6	7

(c)

15	0		12			24	7	6

(d)

15	0		24			12	7	6

(e) none of the above

15	0		12			24	7	6
0	1	2	3	4	5	6	7	8

7 → 7

24%9 = 6 , 24 → 6

6 → 6, oops → 7 oops - 6 → 8

15%9 = 6 , so 15 → 0

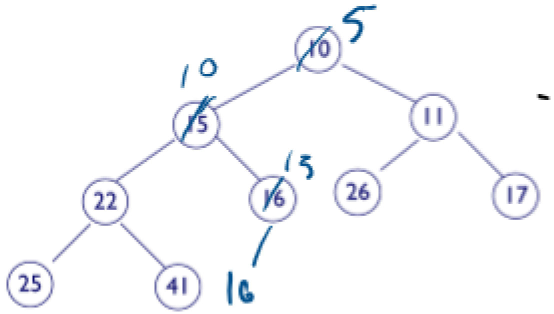
0 → 0 → 1 12%9 = 3

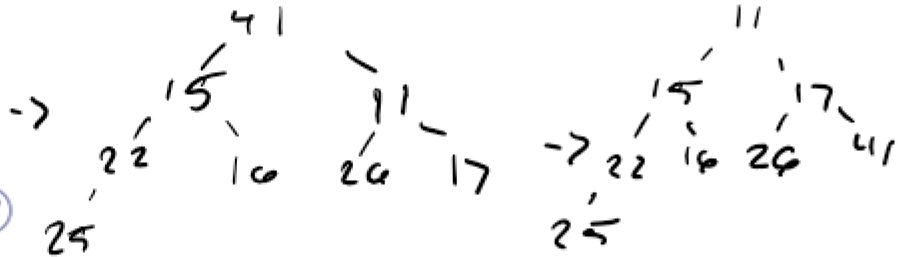
2. Consider a hash table with hash function hash(x) = x % 10 using quadratic probing and a table size of 10. In finding a key of 7, what are the first five locations probed (assuming key 7 is not found)?

(a) 7 8 1 6 3 (b) 7 8 1 2 3
(c) 7 8 9 0 1 (d) 8 9 0 1 2 (e) none of the above

hash(7) = 7 hash(7)+9 = 6
hash(7)+1 = 8 hash(7)+16 = 3
hash(7)+4 = 1 7 8 1 6 3

3. Below is a min-heap stored in an array. Suppose a deleteMin is performed on this heap. What is the new array after the deleteMin operation?





(a) 11 15 17 22 16 26 25 41 (b) 11 15 16 17 22 25 26 41
(c) 11 15 17 22 16 26 41 25 (d) 15 11 17 22 16 26 41 25
(e) none of the above

4. If we insert a new key 5 to the original heap in Question 3, what is the new array after the insertion operation?

(a) 10 5 11 22 15 26 17 25 41 16 (b) 5 10 11 22 15 26 17 25 41 16
(c) 10 15 11 22 5 26 17 25 41 16 (d) 10 15 11 22 16 26 17 25 41 5
(e) none of the above

5. Consider the following array of ten integers: 5 3 8 9 1 7 0 2 6 4. Suppose we partition this array using quicksort's partition function using 5 as the pivot. Which of the following shows a possible array after the partition finishes:

(a) 3 1 0 2 4 5 8 9 7 6 (b) 5 3 4 2 1 0 7 9 6 8
(c) 5 0 3 4 2 1 8 9 7 6 (d) 3 1 0 2 4 8 9 7 6 5 (e) none of the above

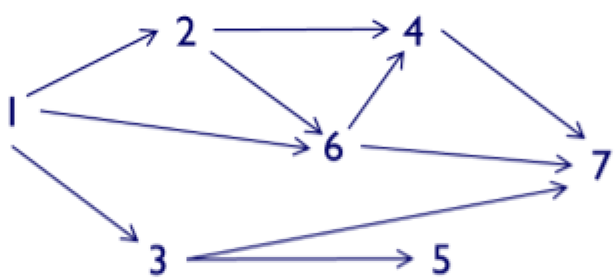
6. We want to sort an array A of n elements. Suppose we know that each element of A is an integer in the range [0, 2n] (i.e., at least 0 and at most 2n). Which is the most efficient algorithm for sorting the numbers in A?

(a) bubble sort (b) heap-sort (c) quick sort
(d) merge sort (e) selection sort (f) insertion sort (g) bucket sort

7. During the depth first search (DFS) on a graph, suppose we are traversing the adjacency list of a vertex u. If we are visiting a vertex v (i.e., we are considering the edge (u,v)), then which color of v indicates that the graph has a cycle?

(a) white (b) blue (c) red (d) none of the above

8. Which of the answers below is NOT a valid topological ordering of the graph?



(a) 1 2 3 6 4 7 5 (b) 1 2 3 4 6 7 5 (c) 1 3 2 5 6 4 7 (d) 1 2 6 4 3 5 7

9. Let G be a directed graph and the weight of every edge is 2. Suppose we want to find a shortest path from a vertex s to a vertex v in G. Which is the most efficient algorithm for finding such a shortest path?

(a) Dijkstra's algorithm (b) breadth first search
(c) depth first search (d) topological sort

10. Let G be an undirected and connected graph, and each edge has a weight. Suppose some edge weights may be negative. If we want to compute a minimum spanning tree of G, does Kruskal's algorithm still work (in particular, due to the negative weights)?

(a) Yes (b) No

Part III. Short Answers (40 points in total).

1. **(5 points)** Using a hash table of size m = 10 with hash function hash(x) = x % 10 and the quadratic probing to resolve collisions, show the hash table that results after the following keys are inserted in the given order: 46 18 27 26 6 20 30.

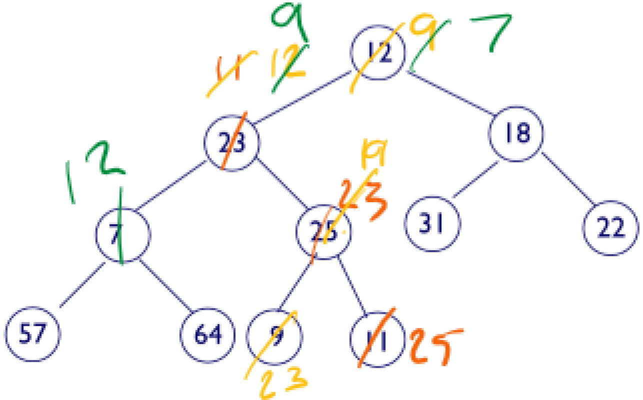
0: 26	1: 20	2:	3:	4: 30	5: 6	6: 46	7: 27	8: 8	9:

46 → 6
18 → 8
27 − 7
26 → 6, 7, 0
6 → 6, 7, 0, 5

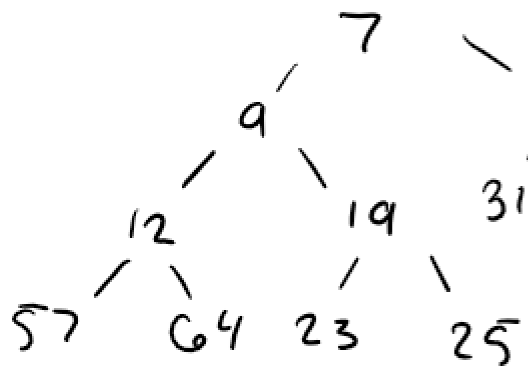
20 → 0 → 1
30 → 0 → 1 → 4

hash
hash + 1
hash + 4
hash + 9
hash + 16

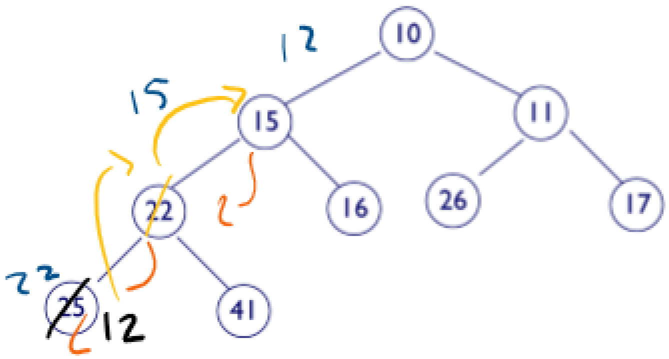
2. **(5 points)** Suppose we use the linear time algorithm to build a min-heap from the following array of numbers: 12 23 18 7 25 31 22 57 64 9 11. The following shows a complete binary tree formed by these numbers. Draw the min-heap after the linear time heap construction algorithm.

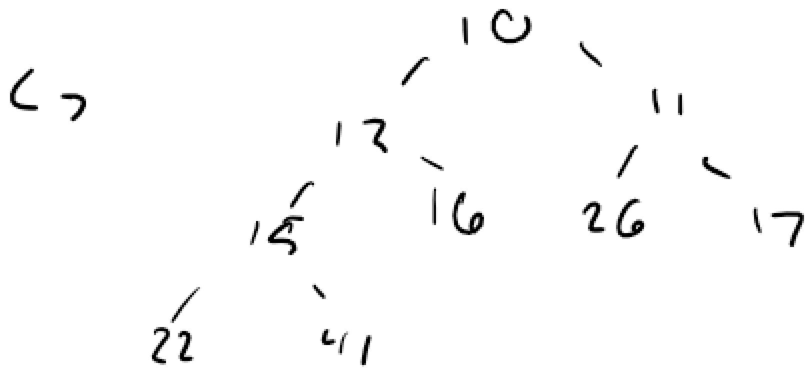


first, second, third

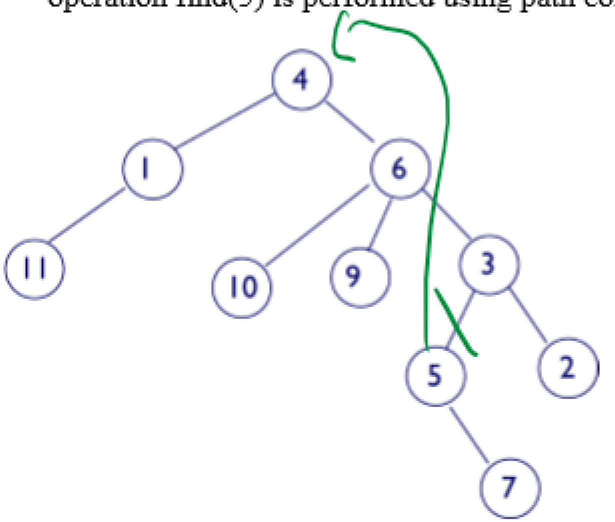


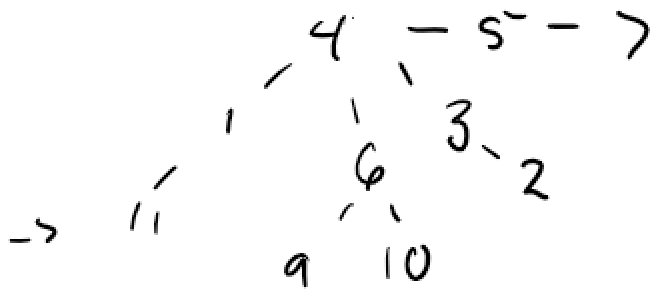
3. **(5 points)** Consider the following min-heap. Suppose we perform a decrease-key operation that decreases the key 25 to 12. Draw the new heap after the above decrease-key operation.



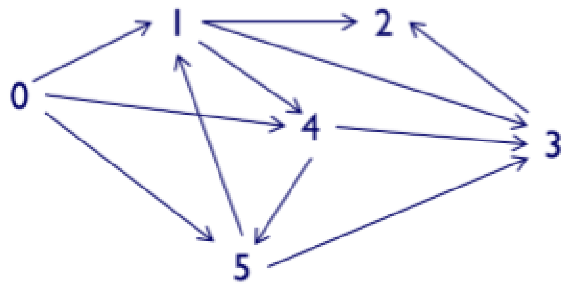


4. **(5 points)** Given the following tree in the disjoint set (union-find) data structure. Draw the new tree after an operation find(5) is performed using path compression.





5. **(10 points)** Consider the following directed graph.



(1) Give the adjacency lists of the graph. For each list, please order the vertices in the ascending order of their indices.

0: 1 -> 4 -> 5

1: 1 -> 3 -> 4

2 :

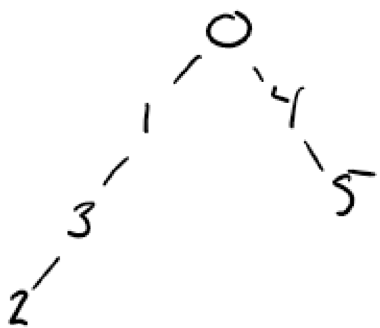
3 : 2

4 : 3 -> 5

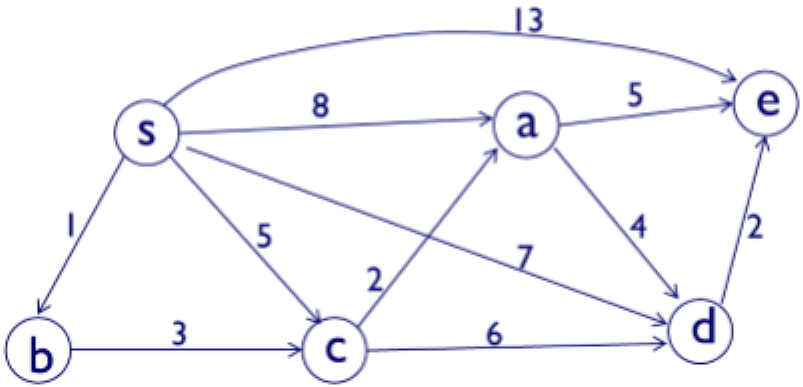
5: 1 -> 3

(2) Suppose we run depth first traversal algorithm on the graph with 0 as the starting vertex and using the adjacency lists you give above. Draw the DFS tree generated by the algorithm.

0 -> 1 -> 3 -> 2 -> 4 -> 5



6. **(10 points)** Step through Dijkstra's algorithm to calculate the single source shortest paths from vertex s to every other vertex in the following graph (the numbers besides the edges are the edge weights).



(1) Draw the shortest path tree produced by the algorithm, and for each vertex, label the shortest path distance from s.

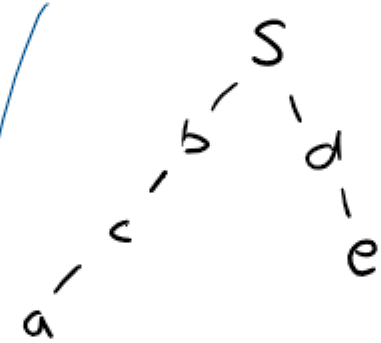
From s to a: s->b->c->a cost 6

s to b: s->b cost 1

s to c: s->b->c cost 4

s to d: s->d cost 7

s to e: s->d->e cost 9



Shortest path tree

(2) Dijkstra's algorithm is implemented by using a priority queue Q. Show the contents of Q just BEFORE vertex a is removed from Q. Specifically, for each vertex v of Q, give the two values d[v] and pre[v], where d[v] is the current shortest path distance from s and pre[v] is the current predecessor of v.

Q = {a, b, c, d, e},

Since I'm assuming a is first. This should be right after the prio queue is made. So, every d[v] would be Integer. Max_Value, and each pre[v] would be -1. The algorith hasn't started yet. It's just about to start.