Final Exam CSCB63 H3Y

Summer 2021

Duration: 8 to 11

Total Points: 120 Marks

Question 1. [0 mark]
Write and then sign the declaration below. When you take a photo of it, place your picture ID beside and hide any sensitive information.

“I have not consulted any resources including but not limited to classmates, tutors, textbooks, webpages, cheat sheets."

Signature: ______________________ Student Number: _________________________

	Number of questions	Marks
Question 1 True or False with Explanation	10	24
Question 2 Short answers	2	8
Question 3 AVL trees and Binary Heap	2	15
Question 4 Graphs	2	20
Question 5 Amortized analysis	1	12
Question 6 Disjoint set and Fibonacci Heap	2	17
Question 7 Hashing	1	12
Question 8 Miscellaneous	1	12

Question 1 [24 Marks]

Descriptions: For following 10 questions, state whether the statements are true or false. Briefly (1 to 3 sentences) justify your answer. Upload a **single file** that contains answers for all the 10 questions.

(a) [2 marks] Polynomial: good. Exponential: bad.
True False
Explain

(b) [2 marks] If the DFS finishing time $f[u] > f[v]$ for two vertices u and v in a directed graph G, and u and v are in the same DFS tree in the DFS forest, then u is an ancestor of v in the depth first tree.
True False
Explain

(c) [2 marks] Let P be a shortest path from some vertex s to some other vertex t in a graph. If the weight of each edge in the graph is increased by one, P will still be a shortest path from s to t.
True False
Explain

(d) [2 marks] If an in-place sorting algorithm is given a sorted array, it will always output an unchanged array.
True False
Explain

(e) [4 marks] Suppose we use a hash function h to hash n distinct keys into an array T of length m. Assuming simple uniform hashing, the expected number of colliding pairs of elements is $\Theta(n^2/m)$.
True False
Explain

(f) [4 marks] The hash family $H = \{h_1, h_2\}$ is universal, where $h_1, h_2 : \{1, 2, 3\} \rightarrow \{0, 1\}$ are defined by the following table:

	1	2	3
h_1	0	1	0
h_2	1	0	1

(For example, $h_1(3) = 0$.)
True False
Explain

(g) [2 marks] Suppose a file server stores a hash of every file in addition to the file contents. When you download a file from the server, you also download the hash and confirm that it matches the file. This system securely verifies that the downloaded file has not been modified by an adversary, provided the hash function has collision resistance.
True False
Explain

(h) [2 marks] The sequence $< 20, 15, 18, 7, 9, 5, 12, 3, 6, 2 >$ is a max-heap.

True False
Explain

(i) [2 marks] Linear probing satisfies the assumption of uniform hashing.
True False
Explain

(j) [2 marks] Dijkstra's algorithm works on any graph without negative weight cycles.
True False
Explain

Question 2. Short answers

Descriptions: For following 2 questions, give short answers. Upload a **single file** that contains answers for all the 2 questions.

(a) [2 marks]
Explain briefly why the statement, "The running time of algorithm A is at least $O(n^2)$," is meaningless.

(b) [6 marks]
You work for the well-known web infrastructure company *Wawaiki*. Your company's business is caching the files associated with requested URLs so browsers can load pages faster. Recent analysis of your caching logs show that more than 3/4 of requested URLs are never requested again.

You're trying to figure out a way to avoid caching the files associated with URLs until the second time you see the URL. Answer following questions based on bloom filter:

i. Briefly explain the reason of using bloom filters for this application and describe the how insertion, deletion, and search of URLs is done with bloom filter.
ii. How will you decide the number of hash functions to use for this application? (Briefly describe the factors you will consider).
iii. What happens to the false positive rate as users request more and more URLs? How should we resolve this problem?

Questions 3. AVL tree, binary heap

Descriptions: For following 2 questions(both (a) and (b)), upload a **single file** that contains answers for all the 2 questions.

(a) [10 marks]
Given the following Integer elements:
`6, 3, 2, 10, 8, 1, 11, 7, 5, 4, 9`
Draw the tree that results when all the above elements are added (in the given order) to each of the following initially empty data structures:

a. An **AVL tree**. (No removals are performed here.) Show the rotations.
b. A minimum binary **heap**. After adding all the elements, perform **3 removes** on the heap.

Please show your work. You do not have to draw an entirely new tree after each element is added or removed, but since the final answer depends on every add/remove being done correctly, you may wish to show the tree at various important stages to help earn partial credit in case of an error.

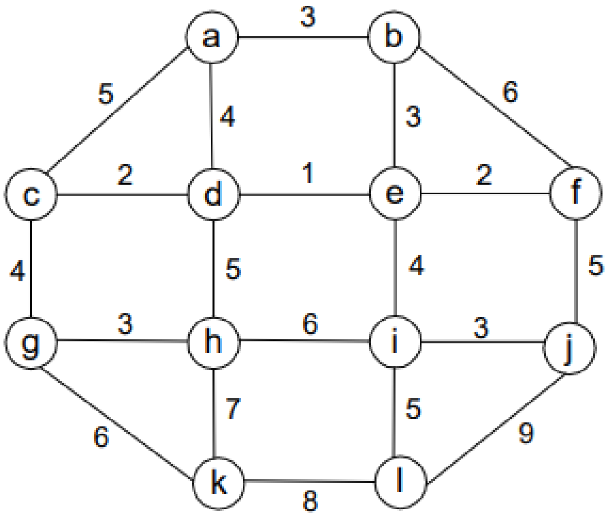
(b) [5 marks]
Suppose you have a **min-heap** H. Give an algorithm that, given a value x, not necessarily in the heap, outputs all elements in H with value less than x. Your algorithm must not modify the heap and must run in $O(k)$ time, where k is the rank of x among the elements in H (i.e., x would be the kth smallest key).
Briefly justify why your algorithm is correct and runs in $O(k)$ time.

Question 4. Graphs

Descriptions: For following 2 questions(both (a) and (b)), upload a **single file** that contains answers for all the 2 questions.

(a) [10 marks]
Minimum Spanning Tree
Let $G = (V, E)$ be a connected, undirected graph with edge-weight function $w : E \rightarrow \mathbb{R}$, and assume all edge weights are distinct. Consider a cycle $\langle v_1, v_2, \ldots, v_k, v_{k+1} \rangle$ in G, where $v_{k+1} = v_1$, and let $(v_i,\ v_{i+1})$ be the edge in the cycle with the largest edge weight.
Prove that $(v_i,\ v_{i+1})$ **does *not* belong** to the minimum spanning tree T of G.

(b) [10 marks]
Apply **Prim's algorithm** to the following graph to obtain minimum spanning tree.
Include in the priority queue only the fringe vertices (the vertices not in the current tree which are adjacent to at least one tree vertex).



Question 5. Amortized analysis

Descriptions: For following question, upload a **single file** that contains answer to this question.

[12 marks]

An array $A[0\,..\,k\ -1]$ of bits (each array element is 0 or 1) stores a binary number
$x\ =\ \sum_{i=0}^{k-1} A[i].\,2^i$
To add $1\ (modulo\ 2^k)$ to x, we use the following procedure:

```
INCREMENT(A, k)
**while** $i < k$ and $A[i] = 1$
```

```
INCREMENT(A, k)
$i \leftarrow 0$
while $i < k$ and $A[i] = 1$
     do $A[i] \leftarrow 0$
          $i \leftarrow i + 1$
if $i < k$
     then $A[i] \leftarrow 1$
```

Given a number x, define the potential $\Phi(x)$ of x to be the number of 1's in the binary representation of x. For example, $\Phi(19) = 3$, because $19\ =\ 10011_2$.
Use a **potential-function argument** to prove that the amortized cost of an increment is $O(1)$, where the initial value in the counter is $x\ =\ 0$.

Question 6. disjoint set, Fibonacci heap

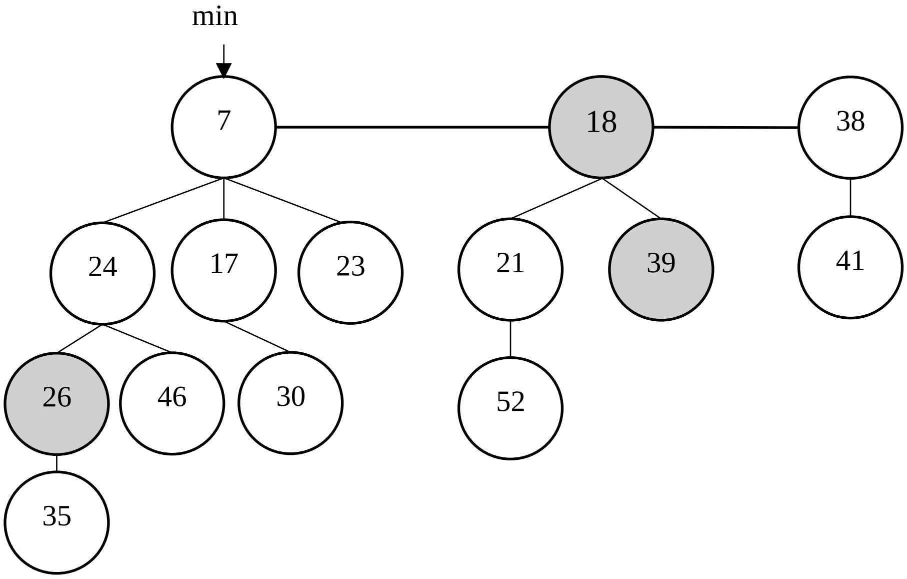
Descriptions: For following 2 questions(both (a) and (b)), upload a **single file** that contains answers for all the 2 questions.

(a) [12 marks]
Let T_1 and T_2 be tree implementations of the disjoint set ADT. T_1 uses union-by-rank **without** path compression, and T_2 uses union-by-rank **with** path compression.
We say two collections of trees are "*equivalent*" if they store the same items and can be drawn the same. For example, the trees T_1 and T_2 constructed from the sequence $MAKE-SET(1), MAKESET(2), UNION(1,2), FIND-SET(1)$ are the same.
Consider a sequence of $MAKE-SET, FIND-SET$, and $UNION$ operations $P = P_1, P_2, \ldots\ P_n$, and suppose that we perform P on each of T_1 and T_2 .
Assume for the $UNION(x, y)$ operation, when there is a choice, that the representative of the set that contains x becomes the new representative.

What is the **smallest number $\boldsymbol{n}$** such that T_1 and T_2 are not equivalent after executing P on each one?
Justify why there is not a smaller n.

(b) [5 marks]

Show the Fibonacci heap that results from calling **FIB-HEAP-EXTRACT-MIN** on the Fibonacci heap shown in Figure. Show your work to receive partial credit.



Question 7. Hashing

Descriptions: For following question, upload a **single file** that contains answer.

[12 marks]

Simulate the behavior of a hash set storing integers given the following conditions:

- The initial array length is 5.
- The set uses quadratic probing for collision resolution.
- The hash function uses the int value (plus any probing needed) mod the size of the table.
- When the load factor reaches or exceeds 0.5, the table enlarges to double capacity and rehashes values stored at smaller indices first.
- An insertion fails if more than half of the buckets are tried.

Recall quadratic probing formula from lecture

$s_i = (h(k) + c_1 i + c_2 i^2) \, mod \, m \qquad for \; i = 0, 1, 2, \ldots .$

and assume c_1 = 0 and c_2 = 1.

Given the following code:

```
Set <integer> set = new HashSet <integer> (5);
set.add(86);
set.add(76);
set.add(16);
set.add(66);
set.add(26);
```

i. What are the indices of the values in the final hash set? Show your work to receive partial credit.

ii. What is the size of the final set?

iii. What is the capacity of the final set?

iv. What is the load factor of the final set?

Question 8. Miscellaneous

Descriptions: For following question, upload a **single file** that contains answer to this question.

[12 marks]
Design a data structure that maintains a dynamic set S of n elements subject to the following operations and time bounds:

Operation	**Effect**	**Time Bound**
1. INSERT(x, S)	Insert x into S.	$O(log\ n)$ expected
2. DELETE(x, S)	Delete x from S.	$O(log\ n)$ expected
3. SUCCESSOR(x, S)	Find the smallest element in S larger than x.	$O(log\ n)$ worst-case
4. FIND-MIN(S)	Return the smallest element in S.	$O(1)$ worst-case
5. SEARCH(x, S)	Return TRUE if element x is in S.	$O(1)$ expected

Describe how the operations are implemented on your data structure and **justify their runtime**.