Montana State University Computer Science Department

Graduating Senior Exam

Spring Semester 2022, 80 minutes.

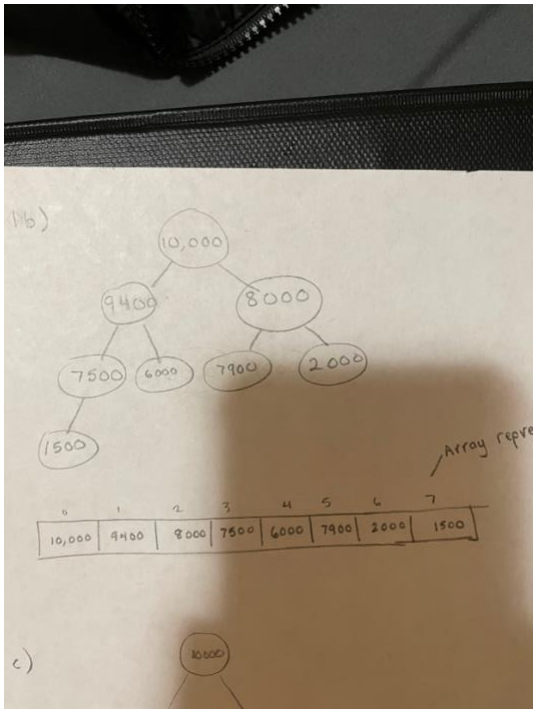
Your performance helps us assess how well we are meeting our stated learning outcomes for computer science majors. To pass this exam, spend at least 45 minutes in order to provide a carefully reasoned answer to each question. This information will help us make improvements.

Name Joaquin Monterrosa

Question 1: A company uses a ticket system for customer service. Each ticket is given an urgency score when it is entered into the system. For example, a VIP customer would receive a higher urgency score than a regular customer. Assume that the urgency scores range from 1 to 10,000. The company's customer service team needs a way to find the most urgent ticket in the system and remove it when it has been addressed. New tickets can arrive at any time.

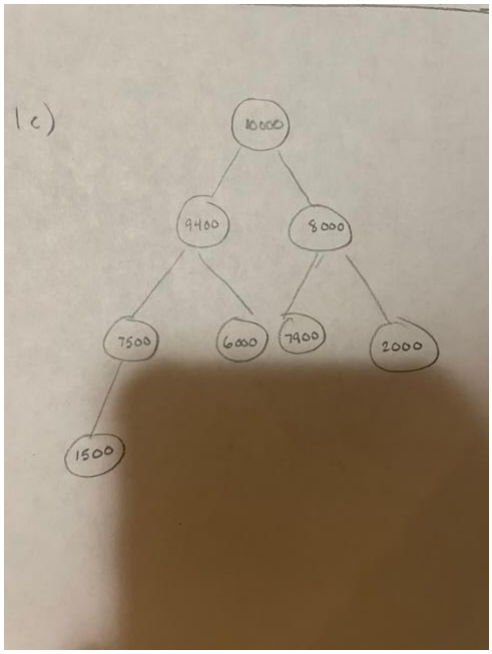
a) What simple data structure would best model this situation? Explain.
Assuming 10,000 signifies the highest urgency and 1 signifies the lowest urgency, the data structure that would best model this situation is a max-heap. A max-heap is the best data structure because the root node will always be the ticket with the highest urgency score so it is perfect for this priority system. Additionally, we can add new tickets efficiently. This data structure is about as efficient as it gets in terms of space and time. Inserts and extracts are O(log n).

b) Describe one way in which this data structure could be implemented and draw a picture of that data structure.
This data structure can be implemented using an array. An array can represent a heap by storing the nodes of the array in level order (starting from the top) going left to right. For example, the root node would be in the 0th index, then its left child would be in the 1st index, then its right child would be in the 2nd index, and so on.



c) Describe a second way in which this data structure could be implemented and draw a picture of that data structure.

A heap can also be implemented with a tree structure.



Question 2: Write a function in a language of your choice that has two parameters: a string and a character. The function should return the number of times that the character occurs in the string.

Your language of choice: Java

Your solution:

```
	public static int letterOccurances(String theString, char theChar) {
		int occurances = 0;

		for(int i = 0; i < theString.length(); i++) {
			if (theString.charAt(i) == theChar) {
				occurances++;
			}
		}

		return occurances;
	}
```

Question 3: Consider a singly linked list that contains **n** floating point numbers and is maintained in ascending order. An additional **k** floating point numbers will be added to the list, one at a time. You may assume that **k** << **n**.

Best case time complexity of algorithm: **Ω**(k)

Explanation of best-case time complexity:
The best case time complexity will happen if the k numbers that are added to the list are sequentially smaller than the smallest floating point number in the singly linked list. In this situation, each number that is added to the list only has to look at the head of the list and realize that that number is bigger than the number being added so it can just be added as the new head of the list. Essentially, each new number will just be added to the head of the list without having to do any traversals.

Worst case time complexity of algorithm: O(k*n) or O(k* n * (k(k+1))/2) to be exact

Explanation of worst-case time complexity:
The worst-case time complexity will happen if the k numbers that are added to the list are sequentially larger than the largest floating point number in the singly linked list. In this situation, each number being added will have to traverse the entire list before it is added as the tail of the list.

Question 4: List any extracurricular community involvement (e.g. the Gallatin Valley Food Bank or coaching), extracurricular university involvement (e.g. Engineering Ambassadors, Swing Cats, AWC or ACM) or paid internships (e.g. Workiva intern or Undergraduate Course Assistant) that you were involved with during your time at MSU. For each activity, (1) **provide** the name of the organization, (2) **describe** briefly your main activities and (3) **estimate** the total number of hours you spent.

Organization	Main Activities	Hours Spent
NASA – Software Engineer	I worked on the Kennedy Space Center Launch Control Systems. I strengthened and expedited their hardware configuration update Procedures and I updated their launch dashboard. Additionally I developed automated tests.	800 hours
Moss Adams – IT Analyst	I audited software systems to ensure they met certain compliance standards (SOC 1, SOC 2, PCI, etc).	480 hours
Freats LLC – Business Strategist and Co-Founder	Freats was a food sustainability app that connected students with leftover food from catered events on campus.	1,000 hours
Legitimizer – Developer and Co-Founder	Legitimizer was a mobile cybersecurity app. It analyzed the Google Play Store and flagged illegitimate apps or reviews based on a variety of heuristics.	500 hours
Blackstone Launchpad - Venture Coach	I helped early-stage entrepreneurs turn their business ideas into reality and I consulted later-stage businesses on best practices.	1,000 hours
Pi Kappa Alpha – Vice President	I was in charge of managing a 110-man chapter.	400 hours
Jake Jabs Student Ambassador	I showed prospective jabs students around the college and mentored freshmen business students	80 hours
LaunchCats Entrepreneurship Club - President	I was in charge of managing the entrepreneurship club, organizing events, and doing design thinking workshops	80 hours

Question 5: IEEE P7003 defines the Algorithmic Bias Standard.

a) Describe briefly what algorithmic bias is.

Algorithmic bias describes unfair judgements made by computer systems. Just like humans make unfair judgements based on protected characteristics like race, gender, or sexuality, computers do too. This could be due to how the system is programmed, what data the system is trained on, or what the software fails to account for.

b) Provide a hypothetical example of how algorithmic bias could manifest itself in college admissions software.

College admissions software is very nuanced and there are many places where algorithmic bias could be introduced but one of the ways the system could become biased is by not factoring in the availability of AP classes at student's high-schools into the algorithm. For example, if a dedicated but underprivileged student attends a high-school where there are fewer resources and fewer AP classes available then that students weighted GPA will naturally be lower than another students weighted GPA simply because of the availability of AP classes. This context is very hard for software to account for and this lack of contextual understanding can lead to algorithmic bias.

Question 6: In 2010 Oracle sued Google claiming copyright and patent infringement for 11,500 lines of Java API copied code. Google used this small subset of code from the JAVA API for their mobile platform operating system. Pick a side (Google or Oracle) to win this lawsuit and explain 3 possible impacts to the computing industry of this win.

Side you choose to win: If Oracle won

First possible impact: It would negatively impact the community-based culture of computer science. Lots of innovation in the world of computer science relies on the ability to "stand on the shoulders of giants". Very few projects have been written 100% from scratch. It's fun and good for the community to be able to reuse snippets of code. If Oracle won, programmers would likely be more hesitant to copy snippets of other people's code.

Second possible impact: It would force software companies to do more meticulous code reviews. Many developers use code from other sources often without thinking twice so software companies would have to do more code inspection to ensure that what happened to Google doesn't happen to them.

Third possible impact: It would disincentivize programmers from using Java for commercial projects. Once people see how aggressive Oracle can be and they understand that Java is owned by Oracle, they might not want to use Java for their projects.

Question 7: In the ACM Code of Ethics and Professional Conduct, one ethical principle stated is to ***avoid harm***. Describe three different types of harm that the software of an autonomous, self-driving car might cause.

First type of harm: Physical Injury - If the software for a self-driving car fails and the car crashes due to that failure, the passengers of that car may sustain injury or even death.

Second type of harm: Property Damage - If the software for a self-driving car fails and the car crashes due to that failure, that car may sustain damage along with any other property involved in the crash.

Third type of harm: Reputation – If the software for a self-driving car fails and the car crashes due to that failure, the company that is producing those autonomous vehicles will likely sustain large reputational damage.

Question 8: Describe your one-year plan following graduation. If you plan to work, list where you have applied (the company and the position) and mention whether you have received any offers. If you plan to go to graduate school, list where you have applied (the school and the area of study) and mention whether you have received any acceptances.

My one-year plan after graduation is to continue working on Astonish Software (my software development agency) and continue working part-time as the technology lead for Posted Software (a job posting startup at MSU). My partner and I have been working on Astonish Software for three months now and have five clients. The goal in the next year is to bring on three more developers and be signing about $20,000 in new contracts monthly. As for Posted, we launched in August 2021 and have a signed a $15,000 contract with MSU. We have 1000+ student users on campus and departments have been vocal about how much the app helps them.

Question 9: List any organized social activities that you were involved with during your time at MSU. For each activity, (1) **provide** the name of the organization, (2) **estimate** the total number of hours you spent.

Organization Name	Hours Spent
Code 24 (annual hackathon at MSU)	100 hours
Cycle For Life (Blood Cancer Fundraiser)	40 hours
Breast Cancer Awareness Merch Fundraiser	40 hours
Intermural Basketball	40 hours

Question 10: List any activities that you were involved with during your time at MSU that prepared you to work more effectively with teammates whose backgrounds are significantly different than your own. For each activity, (1) **describe** the activity and (2) **estimate** the total number of hours you spent.

Activity Description	Hours Spent
I got a business accounting degree	1000000000000000000+ hours
Member of Pi Kappa Alpha	1000 hours
Playing Basketball	200 hours
Attending Honors Presents	50 hours

Question 11: Name the most sophisticated version control system that you use and describe three nontrivial features it has.

Name of most sophisticated version control system: Accurev

Non-trivial feature 1: Resolving Conflicts. Although resolving conflicts is a fairly common task when doing parallel development, it never ceases to be annoying and complicate (especially in Accurev).

Non-trivial feature 2: Automated Testing Integration (Robot Framework). After "promoting" code I would have to test my "stream" (essentially the same thing as a branch) using our Robot Framework automated test suite before I could eventually "update" the main code stream.

Non-trivial feature 3: Jenkins Integrations. I was able to seamlessly integrate Accurev streams with Jenkins' build tools to continuous integration.

Question 12: During your time at MSU, list any major non-classroom, non-paid activities that you engaged in to help prepare for your computer science career. For example, you might have taught yourself a programming language or attended a workshop on a CS topic. For each activity, **estimate** the number of hours spent**.**

Activity Description	**Hours Spent**
Took Codecademy Web Dev Bootcamp	100 hours
React Course	30 hours
Spent the summer of 2020 building a react app	80 hours
Leetcode Challenges	40 hours
Learned Webflow	60 hours
Learned Bubble	200 hours
Learned Blockchain Development	200 hours

Best wishes wrapping up the semester and in the future!