**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 Justin Scarbrough

**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.

The term 'simple data structure' is a little nebulous, so I'm going to assume that it means data structures that don't depend on large or complicated architecture (so something like a hashmap wouldn't be simple). With this in mind, I think a linked list would probably work best for this situation. This is because linked lists can easily be manipulated, and can have entries dynamically removed or added at runtime, which is a part of this question's requirements. Linked list nodes can also hold as much data as they need, so we can hold more than a simple ticketing ID.

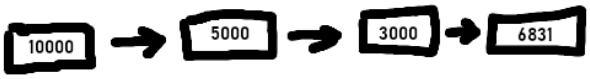
b) Describe one way in which this data structure could be implemented and draw a picture of that data structure.

The first way to implement this data structure would be to set up the linked list to be singly linked and to sort the list after each addition with the highest urgency score being the head node. This will result in a list that can be easily accessed by a customer support team without much searching, but adding new entries might be expensive.

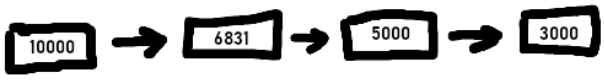
Example List




Adding a Node




Sorted List




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

The second way this list can be implemented is by making a doubly linked list that houses the ticket info and simply remains unsorted. This would result in a more expensive linear search for a highest urgency score, but will be less expensive to add nodes to.

Example List




Adding a Node



**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 (e.g. Python): Python*

*Your solution:*

```
def charCount (word, character):
    count = 0
    for chr in word:
        if chr == character:
            count = count + 1
    return count
```

**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:*

O(k(n))

*Explanation of best case time complexity:*

Regardless of how this is done, we'll always have to sort our list to maintain ascending order, which means we'll be scanning through our list of size n with each addition. In a best case scenario, though, our sorting algorithm will be able to place our new addition as soon as it's done sorting without any other list manipulation. This will happen k times, since this sorting will happen with each entry.

*Worst case time complexity of algorithm:*

$O(k(n^2))$

*Explanation of worst case time complexity:*

What will likely happen is that our linked list will have to scan through each of its values, then sort the list appropriately to accommodate the new addition. I'm assuming we're using a reasonably inefficient sorting algorithm (like bubble sort) for this algorithm, which has a worst case time complexity of $n^2$. In this case, our algorithm will have iterated through our list twice in a worst case

scenario before successfully sorting our new entry. This will happen k times, since the list maintains ascending order after each entry.

**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 (estimate)* |
|---|---|---|
| AFROTC | Event Security (Football games) | ~60 |
| AFROTC | Drill & Ceremony (Colorguard & Drill Team) | ~30 |
| MSU | Undergrad TA Course (CSCI 495, compensated with credit hours) | ~48 |
| | | |
| | | |
| | | |
| | | |
| | | |

**Question 5:** IEEE P7003 defines the Algorithmic Bias Standard.

a) Describe briefly what algorithmic bias is.

In essence, algorithmic bias is the (intentional or unintentional) human bias that someone puts into their code when making an algorithm, usually with something complicated like robot vision.

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

This is probably a bit of a fictitious example, but one algorithm for college admissions could be sorting essays, with algorithmic bias being to score essays with less sophisticated vernacular lower than those with a better one. This could have impacts on certain socioeconomic groups that have less time to study vocabulary, but might still have very good ideas to convey with more basic words.

**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:*

Oracle

*First possible impact:*

Large portions of code (11,500 is large in this case) might be strictly subject to patents and copyright beyond 'intellectual property' (similar to written works of art or technical papers).

*Second possible impact:*

FOSS (Free Open Source Software) might become more widely used by companies due to the lack of legal hurdles and risk of using proprietary code.

*Third possible impact:*

FOSS might become less prevalent due to tighter rules to copyright in code (and therefore more incentive to make software proprietary).

**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 harm through crashes that could have been avoided or mitigated.

*Second type of harm:*

Immediate mental harm through trauma related to crashes and other accidents unique to autonomous cars.

*Third type of harm:*

Chronic mental harm (and financial harm) to consumers. Autonomous cars will be expensive for the first few years of their use, and will likely need universal adoption to work properly.

**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.

I plan on trying to join the US Air Force in may 2023, which is the next selection cycle for officers. In the meantime, I'm planning to work locally. Currently I've applied to IT Positions with two companies (Advicepay and Murdoch's Home & Ranch), as well as a web-developer position with Conveyabull. I've yet to be offered any position as of April 12, 2022.

**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 (estimate)* |
|---|---|
| Sigma Nu Fraternity | ~15 |
| Bozeman Navigators | ~100 |
| | |
| | |
| | |
| | |
| | |
| | |

**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 (estimate)* |
|---|---|
| EGEN 310 Coursework<br>(Iterative design and engineering with non-CS majors) | ~100 |
| Hobbyist Game Design, implementing (in code) ideas and concepts that others want. | ~500 (likely many more) |
| | |
| | |
| | |
| | |
| | |
| | |

**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:*

GitHub

*Non-trivial feature 1:*

Merging commits from people while also pushing changes in the same file.

*Non-trivial feature 2:*

Creating and manipulating folders of a repo on Github's website.

*Non-trivial feature 3:*

Able to add collaborators with different permissions.

**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 (estimate)* |
|---|---|
| Hobbyist game design (self taught Lua scripting) | ~500 |
| Java Practice (board game interpretations, automation, general practice) | ~200 |
| | |
| | |
| | |
| | |
| | |
| | |

Best wishes wrapping up the semester and in the future!