**Math**

Use the following set definitions for the next several questions:

Let

Let

**1.**

What is the value of x?

**2.**

What is the value of x?

**3.**

What is the value of x?

**4.**

What is the value of C?

What is the value of D?

**Asymptotic Notation**

**5.**

Rate the following asymptotic growth rates in order of speed from fastest (at the top) to slowest (at the bottom):

**6.**

Let be an algorithm with a performance growth rate of .

Let be an algorithm with a performance growth rate of .

Which of these two functions has the faster execution time? Explain your answer.

**7.**

Write a description of the following asymptotic notations:

**8.**

Let be an algorithm with a performance growth rate of .

Let be an algorithm with a performance growth rate of .

You are writing a system and have the choice between the two algorithms above. The first algorithm runs in linear time while the second algorithm runs in logarithmic time. However, the logarithmic algorithm requires a heavy up-front constant-time cost.

Discuss how you would decide which of these two algorithms to use. Is one strictly better?

**Sorting**

**9.**

Sorting algorithms like quicksort and insertion sort are called *comparison sorts*. What is the absolute best-case performance we can hope to get with a comparison sort?

**10.**

Random sort is a form of sorting that takes the entire set and shuffles it randomly, then checks to see if it’s sorted.

What is the best-case running time for this algorithm?

What is the worst-case running time for this algorithm?

Explain your answers.

**11.**

Assume we have three functions:

FIND-UNSORTED(A, value):

This function loops through the array A to find out if the value exists.

FIND-SORTED:

This function performs a binary search on A, which is assumed to be sorted, to find out if the value exists.

QUICKSORT(A):

This function uses quicksort to sort the array.

Now consider the two functions below:

FUNCTION-1(A, value)

return FIND-UNSORTED(A, value)

FUNCTION-2(A, value)

QUICKSORT(A)

return FIND-SORTED(A, value)

Which function will perform faster, FUNCTION-1 or FUNCTION-2? Why?

**Algorithm Analysis**

**12.**

Consider an algorithm that compares two strings.

What is the best-case running time?

What is the worst-case running time?

Explain your answers.

**13.**

Write a function that checks an array for duplicates. It should have the following signature:

bool CheckForDuplicates(int\* pArray, int size);

It should return true if there any duplicate values in the array and false if there aren’t. pArray is the array to check and size is the size of the array.

The algorithm MUST be in memory usage.

The algorithm should be as fast as possible.

You may use the compiler to test your function, but copy & paste the function onto this page.

What is the worst-case running time of the algorithm you wrote?

**14.**

Write a function in C++ that takes in a single parameter. The function should print out every number between 1 and that parameter (inclusive) except for the following conditions:

* If the number is a multiple of 3, print “Fizz” instead of the number.
* If the number is a multiple of 4, print “Buzz” instead of the number.
* If the number is a multiple of both 3 and 4, print “FizzBuzz” instead of the number.

The function signature should look like this:

void FizzBuzz(int maxValue);

You may use the compiler to test your comparison, but copy & paste the function onto this page.

**15.**

Write a C++ function that takes a sorted array and a value. The function will search the array and, if the value is found, it will return the index of that value. If the value is not found, it will return the index where this element should be inserted. It should have the following signature:

int FindIndex(const int\* pArray, int size, int value);

For example, consider the following code snippet:

int input[] = {2, 4, 6, 8};

int a = FindIndex(input, 4, 6);

int b = FindIndex(input, 4, 3);

cout << a << “ “ << b;

The output would be “2 1”.

You may use the compiler to test your comparison, but copy & paste the function onto this page.

What is the worst-case running time for this function? Explain your answer.

**Data Structures**

**16.**

The table below has data structures listed on the left and different functions listed across the top. For each cell, fill in the worst-case performance. If a particular operation doesn’t apply to that data structure, put N/A in that cell.

For the purposes of this question, the functions are defined as follows:

Search: Check to see if the data structure has the value you’re looking for.

Index: Check to see if the data structure has the index or key you’re looking for.

Insert: Insert an element at a specific index or key.

Delete: Delete a specific element at an index or key.

Delete Me: Delete a specific element assuming you have the element itself.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Search | Index | Insert | Delete | Delete Me |
| Unsorted Array |  |  |  |  |  |
| Sorted Array |  |  |  |  |  |
| Linked List |  |  |  |  |  |
| Hash |  |  |  |  |  |

**17.**

std::vector has a function called push\_back() that pushes data to the back of the array.

Using asymptotic notation, what is the worst-case run-time performance when calling push\_back()? Explain your answer.

Using asymptotic notation, what is the worst-case memory cost when calling push\_back()? Explain your answer.

**18.**

One key component of hash tables is the hashing function. Write the formal definition of the problem that hashing solves.

For an example of how a formal definition should be written, see the formal definition of the sorting problem below:

**Input:** A sequence of *n* numbers .

**Output:** A permutation (reordering) of the input sequence such that .

This can also be found on page 5 of Introduction to Algorithms.

**19.**

Describe a hash collision.

**20.**

You are tasked with writing a random name generator for a contemporary game. The names come from the US census of all the names given to new babies for the past 20 years. You have three different data sets; the first is for male first names, the second is for female first names, and the third is for surnames. Along with each name is a number, which is how many times that name was used. For example, the male names might look like this:

John, 1034

Michael, 1001

.

.

.

Zaphod, 2

You have the common names at the top with many entries while the uncommon names would be at the bottom. The list itself would likely have hundreds or even thousands of names.

Your first task is to load these names into some kind of data structure. Describe what data structure you would use and why. What operations are you going to need to perform?

**Application of Data Structures & Algorithms**

You are working on a high school simulation game. One of the key components of this game is the relationship system. Each game object has a relationship object that looks like this:

struct Relationship

{

enum class RelationshipType

{

k\_friend,

k\_romantic,

k\_family,

};

// owner and target of the relationship

GameObject\* m\_pOwner;

GameObject\* m\_pTarget;

// how good of a friend we are  
 float m\_relationship;

// the type of relationship this is

RelationshipType m\_type;

// assume appropriate constructor and destructor here

};

Every game object can have any number of relationships, with the following restrictions:

* You may not have a relationship with yourself. In other words, m\_pOwner != m\_pTarget.
* You may not have more than one relationship with the same owner & target. In other words, if there are three students in the world, the maximum number of relationships any student could have is 2 (one for each student that is not you).
* m\_relationship is clamped such that
* If m\_relationship == 0, it means that the students are neutral towards other. Anything < 0 is negative and anything > 0 is positive.

The social game design is as follows:

* Students can talk to each other, which changes the value of their associated relationships.
* The specific socials that students can do with each other is gated by the value of their relationship and the relationship type.
* If a student talks to someone they do not know, an initial relationship object is created.
* While in a conversation, if their relationship score is high enough and they aren’t family members, students have a small chance to express their romantic interest. If this is successful, their relationship type changes to k\_romantic and they will always have the ability to do romantic socials from then on.

Use the above design to answer the questions below.

**21.**

Given the requirements above, what kind of data structure would you use to store relationships on a GameObject? Explain your answer.

**22.**

Using the relationship example above, let’s say that the design team has decided that they want to all students to be monogamous. When a student successfully expresses interest in another student and changes their relationship to be romantic, we need to reset any other romantic relationship to be k\_friends.

Given your answer above, what is the worst-case performance of a successful express interest? Explain your answer.

**23.**

How can you change the architecture so that a successful express interest runs in time AND memory? Explain your answer.

**24.**

Progress continues on the high school simulation game and the designers are adding a new feature. They want you to be able to throw a party and invite all of your friends. The feature works like this:

* Students can throw a party.
* All students with a relationship >= 75 will show up to the party.

Given your answers above, what is the worst-case performance of a function that builds and returns the list of students who should show up? Is there a way we can make it faster without sacrificing the performance/memory costs of the other features? Explain your answers.

**25.**

Designers are enjoying the party feature, but they want to change the threshold for relationship score based on type. Specifically:

* Students who are friends must have a relationship >= 75 to show up.
* Students who are family members must have a relationship > 0 to show up.
* Students who are romantic partners must have a relationship >= 50 to show up.

Does this change the performance characteristics of your answer above? Why or why not?