https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=562504&extra=page%3D5%26filter%3Dsortid%26sortid%3D311%26sortid%3D311

Your instructor comes to you and tells you that they desperately need you to help them write a function for them that can compare two words for alphabetic sorting. They **want** it to return True if the first word comes first alphabetically, and False otherwise.
For example, "a" comes before "b" in the English alphabet, so your instructor's goal is to have compare("a", "b") return True.

Knowing that you're *brilliant*, your instructor would like you to review what they wrote and evaluate what it outputs for a set of inputs. Since they have a great coding style, they've even provided comments trying to explain what they are attempting to do in order to help you out.
You can assume that **all syntax is correct**, that all words are lowercase, and that all strings and lists begin with index 0.

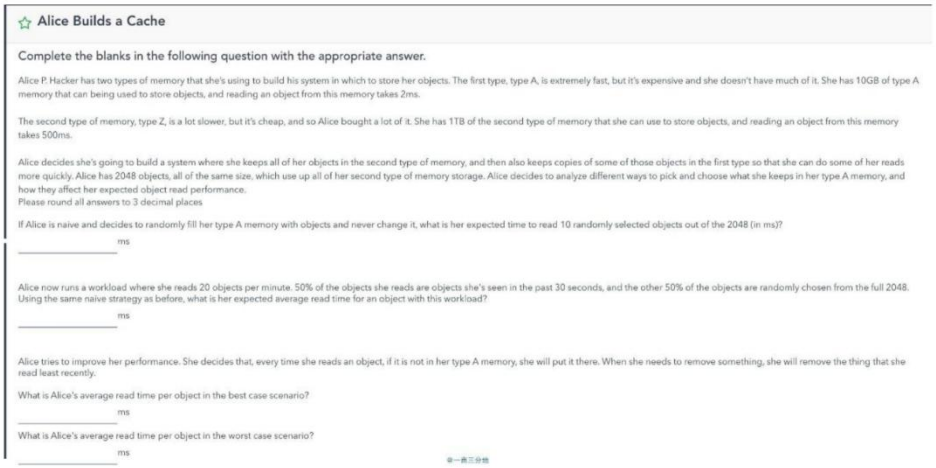
Your instructor shows you the following function, and tells you that "alphabet" is just a list of the alphabet starting at "a" and going all the way to "z":

```
function compare(word1, word2):
    // this should return True if value 1 is alphabetically before value 2, false otherwise
    letter_index = 0
    while letter_index < word1.length()
        letter_index += 1
        letter1 = word1.get_character_at(letter_index)
        letter2 = word2.get_character_at(letter_index)
        if alphabet.indexOf(letter1) > alphabet.indexOf(letter2)
            return True
        else if alphabet.indexOf(letter2) > alphabet.indexOf(letter1)
            return False
    // if they're the same word, we'll exit the loop and it doesn't matter what we return
    return True
```

What will be the output if you use this compare function to sort the list ["fff", "hhh"]? At the end of sorting, compare will return True when comparing the first and second element of the list.

Pick one of the choices

- ["fff", "hhh"]
- ["hhh", "fff"]
- ["hhh", "hhh"]
- ["fff", "fff"]
- **Program will crash, throw an error, or exhibit some other undefined behavior**



Alice Builds a Cache

Complete the blanks in the following question with the appropriate answer.

Alice P. Hacker has two types of memory that she's using to build his system in which to store her objects. The first type, type A, is extremely fast, but it's expensive and she doesn't have much of it. She has 10GB of type A memory that can being used to store objects, and reading an object from this memory takes 2ms.

The second type of memory, type Z, is a lot slower, but it's cheap, and so Alice bought a lot of it. She has 1TB of the second type of memory that she can use to store objects, and reading an object from this memory takes 500ms.

Alice decides she's going to build a system where she keeps all of her objects in the second type of memory, and then also keeps copies of some of those objects in the first type so that she can do some of her reads more quickly. Alice has 2048 objects, all of the same size, which use up all of her second type of memory storage. Alice decides to analyze different ways to pick and choose what she keeps in her type A memory, and how they affect her expected object read performance.
Please round all answers to 3 decimal places

If Alice is naive and decides to randomly fill her type A memory with objects and never change it, what is her expected time to read 10 randomly selected objects out of the 2048 (in ms)?

_____ ms

Alice now runs a workload where she reads 20 objects per minute. 50% of the objects she reads are objects she's seen in the past 30 seconds, and the other 50% of the objects are randomly chosen from the full 2048. Using the same naive strategy as before, what is her expected average read time for an object with this workload?

_____ ms

Alice tries to improve her performance. She decides that, every time she reads an object, if it is not in her type A memory, she will put it there. When she needs to remove something, she will remove the thing that she read least recently.

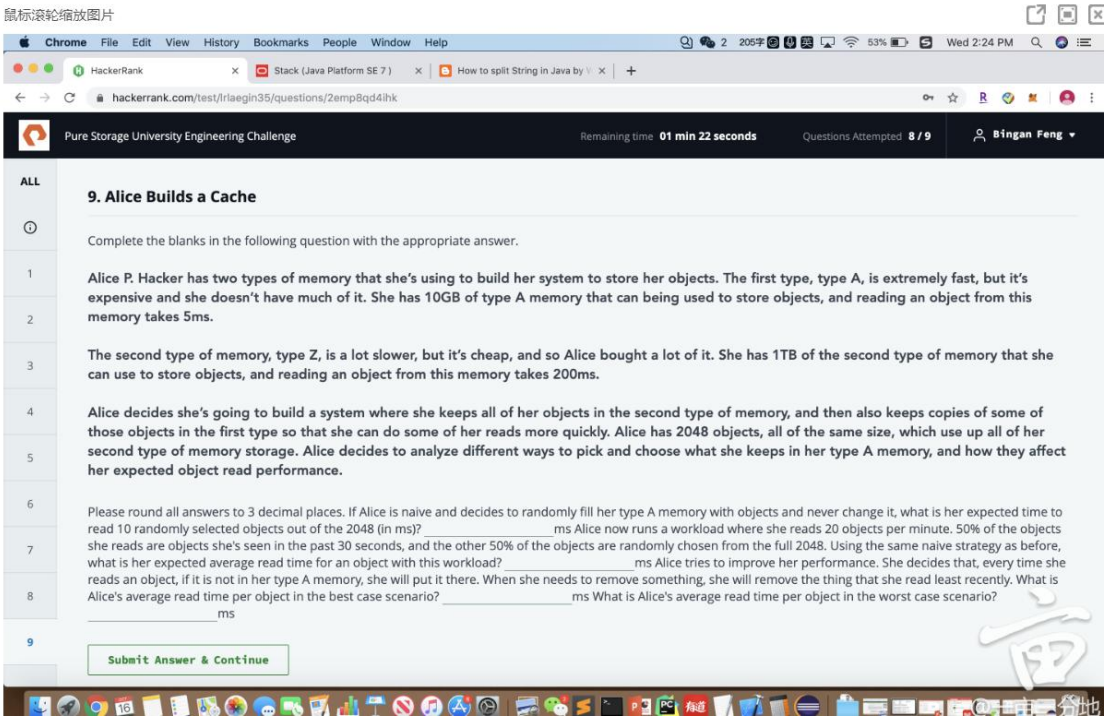
What is Alice's average read time per object in the best case scenario?

_____ ms

What is Alice's average read time per object in the worst case scenario?

_____ ms

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=560787&extra=page%3D19%26filter%3Dsortid%26sortid%3D311%26sortid%3D311



https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=558438&extra=page%3D23%26filter%3Dsortid%26sortid%3D311%26sortid%3D311

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=561456&extra=page%3D23%26filter%3Dsortid%26sortid%3D311%26sortid%3D311

https://www.1point3acres.com/bbs/thread-561456-1-1.html

https://www.1point3acres.com/bbs/thread-565406-1-1.html

We want to store one million "interesting" numbers in a data structure and are choosing between a doubly linked list and an array. The numbers are in the range from 0 to (2^{64}-1), and are generated (and thus stored) in sorted order. If we only consider the numbers stored in our data structure to be interesting, and we are given a random number in the range from 0 to (2^{64}-1), how long does it take for each data structure to determine if a particular number is "interesting"? Assume that it takes 10ns to process a single element of either data structure, and ignore any additional overheads. Use average case performance.

Pick **ONE** option

- (●) It will take us 200 ns for the array and 5 million ns for the linked list.
- ( ) It will take us 200 ns for the array and 10 million ns for the linked list.
- ( ) It will take us 5 million ns for the array and 5 million ns for the linked list.
- ( ) It will take us 5 million ns for the array and 10 million for the linked list.
- ( ) It will take us 5 million ns for the array and 200 ns for the linked list.
- ( ) It will take us 10 million ns for the array and 200 ns for the linked list.

Clear Selection

2. Sorting on a Custom Alphabet

Your instructor comes to you and tells you that they desperately need you to help them write a function for them that can compare two words for alphabetic sorting. They *want* it to return True if the first word comes first alphabetically, and False otherwise.
For example, "a" comes before "b" in the English alphabet, so your instructor's goal is to have **compare("a", "b") return True.**

Knowing that you're *brilliant*, your instructor would like you to review what they wrote and evaluate what it outputs for a set of inputs. Since they have a great coding style, they've even provided comments trying to explain what they are attempting to do in order to help you out.
You can assume that all syntax is correct, that all words are lowercase, and that all strings and lists begin with index 0.

Your instructor shows you the following function, and tells you that "alphabet" is just a list of the alphabet starting at "a" and going all the way to "z":

```
function compare(word1, word2):
   // this should return True if value 1 is alphabetically before value 2, false otherwise
   letter_index = 0
   while letter_index < word1.length()
      letter_index += 1
      letter1 = word1.get_character_at(letter_index)
      letter2 = word2.get_character_at(letter_index)
      if alphabet.indexOf(letter1) > alphabet.indexOf(letter2)
         return True
      else if alphabet.indexOf(letter2) > alphabet.indexOf(letter1)
         return False
   // if they're the same word, we'll exit the loop and it doesn't matter what we return
   return True
```

题干相同

What will be the output if you use this compare function to sort the list ["xxx", "zzz"]? At the end of sorting, compare will return True when comparing the first and second eleme

Pick **ONE** option

- ( ) ["xxx", "zzz"]
- ( ) ["xxx", "xxx"]
- (●) ["zzz", "xxx"]
- ( ) ["zzz", "zzz"]
- ( ) Program will crash, throw an error, or exhibit some other undefined behavior

Clear Selection

What will be the output if you use this compare function to sort the list ["abc", "abcd"]? At the end of sorting, compare will return True when comparing the first and second element of the list.

Pick **ONE** option

- ["abc", "abcd"]
- ["abcd", "abc"]
- ["abcd", "abcd"]
- ["abc", "abc"]
- Program will crash, throw an error, or exhibit some other undefined behavior

Clear Selection

What will be the output if you use this compare function to sort the list ["xyzxyz", "yyyxxx"]? At the end of sorting, compare will return True when comparing the first and second element of the list.

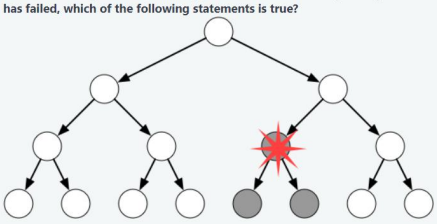
Pick **ONE** option

- ["xyzxyz", "yyyxxx"]
- ["yyyxxx", "xyzxyz"]
- ["yyyxxx", "yyyxxx"]
- ["xyzxyz", "xyzxyz"]
- Program will crash, throw an error, or exhibit some other undefined behavior

Clear Selection

5. Binary Tree Failures

Your system consists of nodes connected in a full binary search tree of height 6. If a node has failed, it is unreachable, as are all of its descendants, as shown in the figure below (the node with the red star has failed). You randomly select a node from the tree, and want to traverse from the root of the tree to your selected node. You are unaware that one node (you don't know which one) has failed, *potentially* causing your traversal to fail, depending on which node you're looking for and which node failed. All nodes are equally likely to fail. Assuming that exactly one node has failed, which of the following statements is true?



Pick **ONE** option

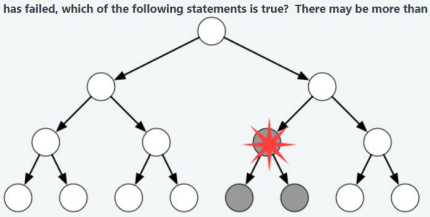
Pick **ONE** option

- The likelihood of success is between 90–100%
- The likelihood of success is between 80–89.99%.
- The likelihood of success is between 70–79.99%.
- The likelihood of success is between 60–69.99%.
- The likelihood of success is between 50–59.99%.
- The likelihood of success is less than 50%.

Clear Selection

6. Binary Tree Failures

Your system consists of nodes connected in a full binary search tree of height 6. If a node has failed, it is unreachable, as are all of its descendants, as shown in the figure below (the node with the red star has failed). You randomly select a node from the tree, and want to traverse from the root of the tree to your selected node. You are unaware that one node (you don't know which one) has failed, *potentially* causing your traversal to fail, depending on which node you're looking for and which node failed. All nodes are equally likely to fail. Assuming that exactly one node has failed, which of the following statements is true? There may be more than one true statement.



Pick **ONE OR MORE** options

- [ ] There is always at least one reachable node.
- [x] The majority of possible failures only make a single node unreachable.
- [x] The likelihood of success increases as the height of the tree increases.

7. Number Scores

You have developed a scoring system for positive integers that works as follows:

- +1 points for every 7 found in the number. For example, 7571 would score 2 points.
- +3 points for each pair of consecutive 5s. If there are more than two 5s in a row, add +3 for each additional 5, since it makes an additional pair (for example, four consecutive 5s gives +15).
- +N^2 points for a sequence of length N ($N >= 1$) where each digit is 1 *less* than the previous digit. For example, 9765320 (9-765-32-0) would be $1 + 3^2 + 2^2 + 1 = 15$ points.
- +2 if the entire number is a multiple of 3
- +4 for each even digit (note that 0 *is* even)

Each component of the score is evaluated separately, so a given digit may contribute to more than one component. For example, the number 765 would score 9 for the sequence of length 3, 4 for one even digit (6), 1 for the 7 digit, and 2 because 765 is a multiple of 3, for a total of 9 + 4 +1 + 2 = 16.

Write a function compute_number_score that computes (and returns) a score for an integer passed to it. The number will be in the range 0<=*number*<1000000000.

8. Lock Use Analyzer

Suppose we want to monitor how locks are used in our system. As the first step, we log moments of acquire and release for each lock in the following format:

- ACQUIRE X
- RELEASE X

where X is some integer ID (1<=X<=1,000,000) of the lock.

All locks must be released in the reverse order of acquiring them; for example, this is a correct event sequence:

1. ACQUIRE 364
2. ACQUIRE 84
3. RELEASE 84
4. ACQUIRE 1337
5. RELEASE 1337
6. RELEASE 364

However, the following sequence violates this rule, because lock 84 is still acquired while releasing lock 364:

1. ACQUIRE 364
2. ACQUIRE 84
3. **RELEASE 364**
4. RELEASE 84

It's also dangerous to leave locks acquired after application termination, as other processes in the system may be blocked while waiting on them, so such sequence is incorrect, too:

1. ACQUIRE 364
2. ACQUIRE 84
3. RELEASE 84

since lock 364 is never released

Third type of problem is lock misuse: it's never good to release a lock that has never been acquired, e.g.:

1. ACQUIRE 364
2. **RELEASE 84**
3. RELEASE 364

and it is as bad to acquire an already acquired lock (usually resulting in a deadlock):

1. ACQUIRE 364
2. ACQUIRE 84
3. **ACQUIRE 364**
4. RELEASE 364

Write a program that, given a list of **N (0<=N<=1,000,000)** lock acquire and release events (counting from 1), checks if there were any problems (acquire-release order violation, dangling acquired lock, acquiring a lock twice or releasing a free lock), and if so, tells the earliest time that could be detected. Note that there's no limit on how many nested locks may be acquired at any given moment.
More formally, you are given an array of strings where each string is either "ACQUIRE X" or "RELEASE X", where all **X**s are integers in the range [1..1000000].
Return:

- **0**, if there were no lock-related problems even after program termination
- **N+1**, if the only issue after program termination were dangling acquired locks
- **K**, in case event number **K** violated any of the principles (release a lock not acquired previously, acquire an already held lock OR violate lock acquire-release ordering).

Examples:
Input:

1. ACQUIRE 364
2. ACQUIRE 84
3. RELEASE 84
4. RELEASE 364

Output: 0 (nothing bad happened)

Input:

Input:

1. ACQUIRE 364
2. ACQUIRE 84
3. RELEASE 364
4. RELEASE 84

Output: 3 (lock 84 should have been released before releasing 364)

Input:

1. ACQUIRE 123
2. ACQUIRE 364
3. ACQUIRE 84
4. RELEASE 84
5. RELEASE 364
6. ACQUIRE 456

Output: 7 (upon terminating, not all locks were released, namely 123 and 456, but we can't know that until actually exiting)

Input:

1. ACQUIRE 123
2. ACQUIRE 364
3. ACQUIRE 84
4. RELEASE 84
5. RELEASE 364

Output: 7 (releasing a lock not acquired before)

Input:

1. ACQUIRE 364
2. ACQUIRE 84
3. ACQUIRE 364
4. RELEASE 364

Output: 3 (acquiring an already held lock)

9. Alice Builds a Cache

Complete the blanks in the following question with the appropriate answer.

Alice P. Hacker has two types of memory that she's using to build his system in which to store her objects. The first type, type A, is extremely fast, but it's expensive and she doesn't have much of it. She has 10GB of type A memory that can being used to store objects, and reading an object from this memory takes 2ms.

The second type of memory, type Z, is a lot slower, but it's cheap, and so Alice bought a lot of it. She has 1TB of the second type of memory that she can use to store objects, and reading an object from this memory takes 500ms.

Alice decides she's going to build a system where she keeps all of her objects in the second type of memory, and then also keeps copies of some of those objects in the first type so that she can do some of her reads more quickly. Alice has 2048 objects, all of the same size, which use up all of her second type of memory storage. Alice decides to analyze different ways to pick and choose what she keeps in her type A memory, and how they affect her expected object read performance.

Please round all answers to 3 decimal places If Alice is naive and decides to randomly fill her type A memory with objects and never change it, what is her expected time to read 10 randomly selected objects out of the 2048 (in ms)? 4951.367 ms Alice now runs a workload where she reads 20 objects per minute. 50% of the objects she reads are objects she's seen in the past 30 seconds, and the other 50% of the objects are randomly chosen from the full 2048. Using the same naive strategy as before, what is her expected average read time for an object with this workload? 495.137 ms Alice tries to improve her performance. She decides that, every time she reads an object, if it is not in her type A memory, she will put it there. When she needs to remove something, she will remove the thing that she read least recently. What is Alice's average read time per object in the best case scenario? 2 ms What is Alice's average read time per object in the worst case scenario? 500 ms

Submit Answer & Continue