Parta

1. What did you implement as your first hash function? How does it work? (No more than 100 words)

文本

描述已自动生成

**First, I create a new hash table to store the string value I need to look up. Secondly, I created a hash function to convert the string into a hash value, and then use the hash value to calculate the hash code, which is used as the index of the hash table.**

**In this way, we can smoothly store the string value in the hash table to ensure that it is convenient and faster to find him.**

1. How do find and insert with linear probing work? Illustrate your explanation with a sample of your code for insert - this may be tidied up a bit for presentation but should clearly be the same as the code you have submitted. (no more than 200 words, not counting the code snippet).

文本

描述已自动生成

**For the insert function, we first need to successfully put the string value into the hash table. When we have a new string value that we want to add to the hash table, we need to check if it already exists in the hash table. If yes, there is a collision, the collision count plus 1 and false is returned to indicate that the insert failed. If the string value does not exist in the hash table, we convert it into a hash value and calculate its hash code and insert it into the hash table. After insert successfully, return True.**

文本

描述已自动生成

**For the find function, we first need to successfully convert the string value into a hash value, and calculate its hash code for the index of the hash table. With this value, we compare it with the one that exists in the hash table. If they are equal, return true, otherwise return false.**

1. What did you implement as your second hash function? How does it work? (No more than 100 words)
2. Did you implement rehashing? If so briefly explain your implementation including explaining when you choose to rehash. (No more than 200 words, you may include code snippets which do not count towards the word count)

python3 test.py python simple hashset 2

python3 test.py python simple bstree 2

partb1. How do find and insert for binary trees work? Include your code for insert and relate your explanation to this code. (No more than 200 words, not including the code snippet)

partc1. Does your implementation of insert for hash sets work as expected in terms of complexity. You

should consider only one of your hash functions, with linear probing.

2. Does your implementation of insert for binary trees work as expected in terms of complexity.

This should consider both average and worst case.

要求Hypothesis You should state, as a hypothesis, what you expect the performance to be and then

write a short paragraph explaining why you believe this to be the case, based on the theoretical

complexities of insert for the data structure. (This should be about 200 words and take up no

more than half a page in your report).

Design You should describe how you designed the experiment. This should include:

• The input files you used and why;

• How you reduced the chance that some randomly generated input had properties that were

not typical of the average case;

• The command line call(s) you used or, if you created your own script for this, include the

script as an appendix or give its name and include it in the code you submit to GitLab and

the zip file you upload to Blackboard;

• Anything else that would help your marker judge how well you designed your experiment

to test your hypothesis.

(This should be about 500 words and take up no more than a page in your report – not

counting any scripts included as appendices).

Results You should present your results as a graph or a table. If you do any processing on results,

such as generating best fit lines, computing averages, etc., then these should be described. Raw

data should be presented in an appendix or included with your code submission, if any processing

has taken place here. (This should be about 200 words and take up no more than half a page

in your report – not counting graphs and tables).

Discussion You should briefly discuss whether your results confirm your hypothesis. If they do not

confirm your hypothesis then you should briefly discuss any ideas you have for why they did

not. (This should be about 200 words and take up no more than half a page in your report).

可以不写1. What potential problems might linear probing cause with respect to the clustering of elements in a hash table?

2. How does quadratic probing work and what are its advantages and disadvantages? Include some of your code and relate your explanation to specific lines in the code. (no more than 200 words not counting code snippets)

3. How does double hashing work and what are its advantages and disadvantages? Include some of your code and relate your explanation to specific lines in the code. (no more than 200 words not counting code snippets)

4. How does separate chaining work and what are its advantages and disadvantages? Include some of your code and relate your explanation to specific lines in the code. (no more than 200 words not counting code snippets)