Drew Smith

12/2/20

CS 124

Project 5

My data set is the results of all the professional tennis tournaments that happened in the year 2000. The six item from this data set that I am representing in my class are the name of the tournament, the name of the match winner, the level rating of the match, the age of the winner, the score of the sets played, and my unique Identifier is created using a combo of the date the matches happened as well as the match number. This made comparing my data slightly more difficult I ended up using some code I found on stack overflow to help me break this data apart into the most important parts. Simply the date and the match number this is due to the year always being 2000. A game is considered greater than another if its match number is higher or if it is later into the year.

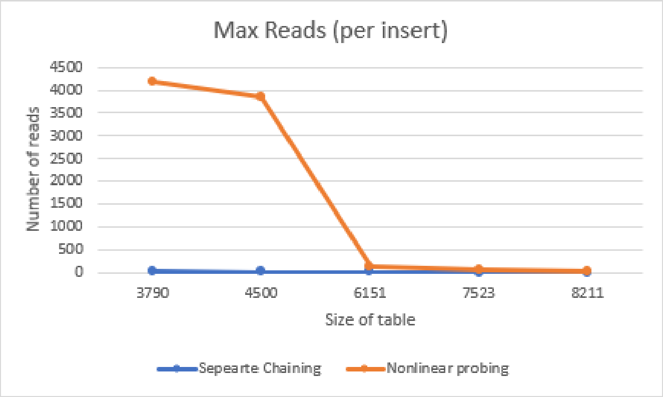
- Tournoment (string type, name of the tournament the matches are in)
- Winner (string type, the name of the winner of the particular match)
- Level (string type, the level rating of the tournament the matches are in)
- Age (float type, the age of the players is saved as a float)
- Score (string type, the score from as many as 5 matches to determine the winner)
- uniquiIdentifer (string type, combo of the date and match number)

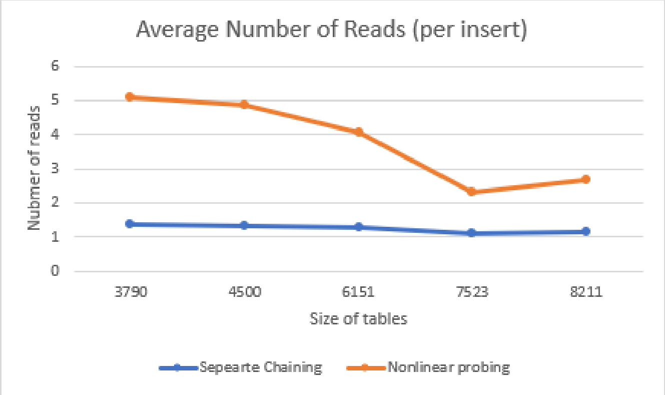
I chose this data set because I have always been interested in sports and tennis was a sport I new very little about so I saw this as not just an opportunity to learn more about computer science but to expand my understanding of a sport I knew very little about.

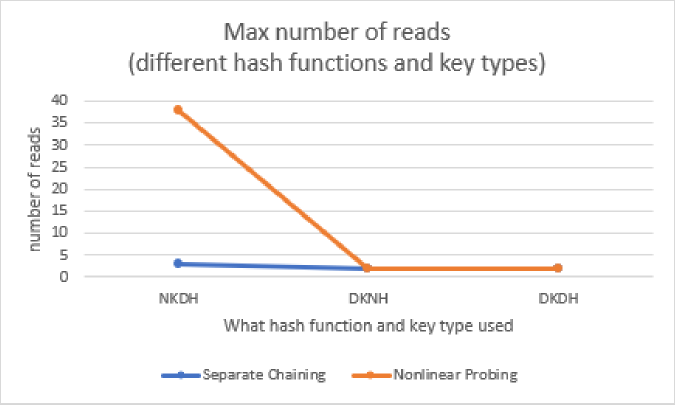
For part B of this project with data collection I was able to determine that for my data set and object type the fastest table size for me to use was my 4th table size being 7523. While the max number of reads does not show this. In both versions the Nonlinear probing and Separate chaining the average number of reads is lower than both the smaller tables and the larger tables.

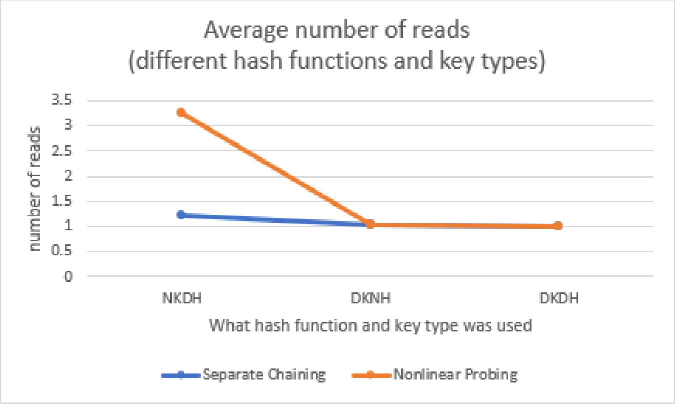
For part C in this lab we were supposed to make a new version of the hash function and use a different key function. I accomplished this by copying the class and changing the hash function this was simply an easy way to make all of the previous parts of the lab work with out having to fun the code in segments to get all the data. The effect on the placement the change in key had was very significant. To start I used the name of the winner of each match however this is repeated simply too often the number of entries in the table were very far apart. I swapped to using the score of the matches which allowed for less exact matches to come up in the insertion process and the table overall seemed to be a lot better. As for which is the best it would be a tie between the separate chaining and the non linear probing with both a different hash function and a different key.

For part D find compared to inserting would be less reads simply because for insert to work it need to see if there is something there already which means it has to run the find this will in practice lead to more reads then if you were simply trying to find something. For the remove it can lead to more it can also lead to less depends on how far in the item it is trying to find. In the majority of cases it will lead to needing to do more reads then a typical insert.


Max Reads (per insert)
Number of reads
4500
4000
3500
3000
2500
2000
1500
1000
500
0
3790
4500
6151
7523
8211
Size of table
Sepearte Chaining
Nonlinear probing


Average Number of Reads (per insert)
Nubmer of reads
6
5
4
3
2
1
0
3790
4500
6151
7523
8211
Size of tables
Sepearte Chaining
Nonlinear probing


Max number of reads
(different hash functions and key types)
number of reads
40
35
30
25
20
15
10
5
0
NKDH
DKNH
DKDH
What hash function and key type used
Separate Chaining
Nonlinear Probing


Average number of reads
(different hash functions and key types)
number of reads
3.5
3
2.5
2
1.5
1
0.5
0
NKDH
DKNH
DKDH
What hash function and key type was used
Separate Chaining
Nonlinear Probing