Zakaria Almatar

CS163

Program#3

Design Write-up

**1) How well did the data structure selected perform for the assigned application?**

The data structure used was arrays of linked lists with hashing. In hashing by date, the day of the month was used as the key for the hashing function. As a simple design I built it to keep records of one month only and so it had 31 head pointers. So if the user of the application receives more than one call a day chaining would be used. For a person with few calls a day the design would still perform nicely. However if it was used for a phone that receives calls all day (e.g. 911) we will end up with a very long chains and thus hashing won’t be as efficient as it’s suppose to be. Also hashing without chaining for collision resolution would be very inefficient in this situation. The other part of the design was hashing by name which also worked nicely. It allowed more room and thus shorter chains which are more run time efficient.

**2) Would a different data structure work better? If so, which one and why...**

I can’t think of another data structure that would have worked better in this situation. We could though eliminate hashing if we can get the day of the month (1-31) and just use an array of link lists which would have almost similar performance. However, we can’t eliminate hashing when using names so the data structure used fits well with the requirements of the application.

**3) What was efficient about your design and use of the data structure?**

Hashing with name was much more efficient than having to search through a list looking for a matching name.

**4) What was not efficient?**

In the retrieve functions, I had to create a new linear link list and copy the results into it so the user can’t have access to the original information. I think this is efficient for security purposes, but if we were dealing with large pieces of information this would not be as efficient and could possibly be a waste of memory.

**5) What would you do differently if you had more time to solve the problem?**

I’m not sure. Though as an improvement to the design, I would rethink the hashing with date and allow it to take in the month and year and not have it limited to 31 head pointers.