

CSE2031

Winter 2017 Lab 6 Hashing --Revisited

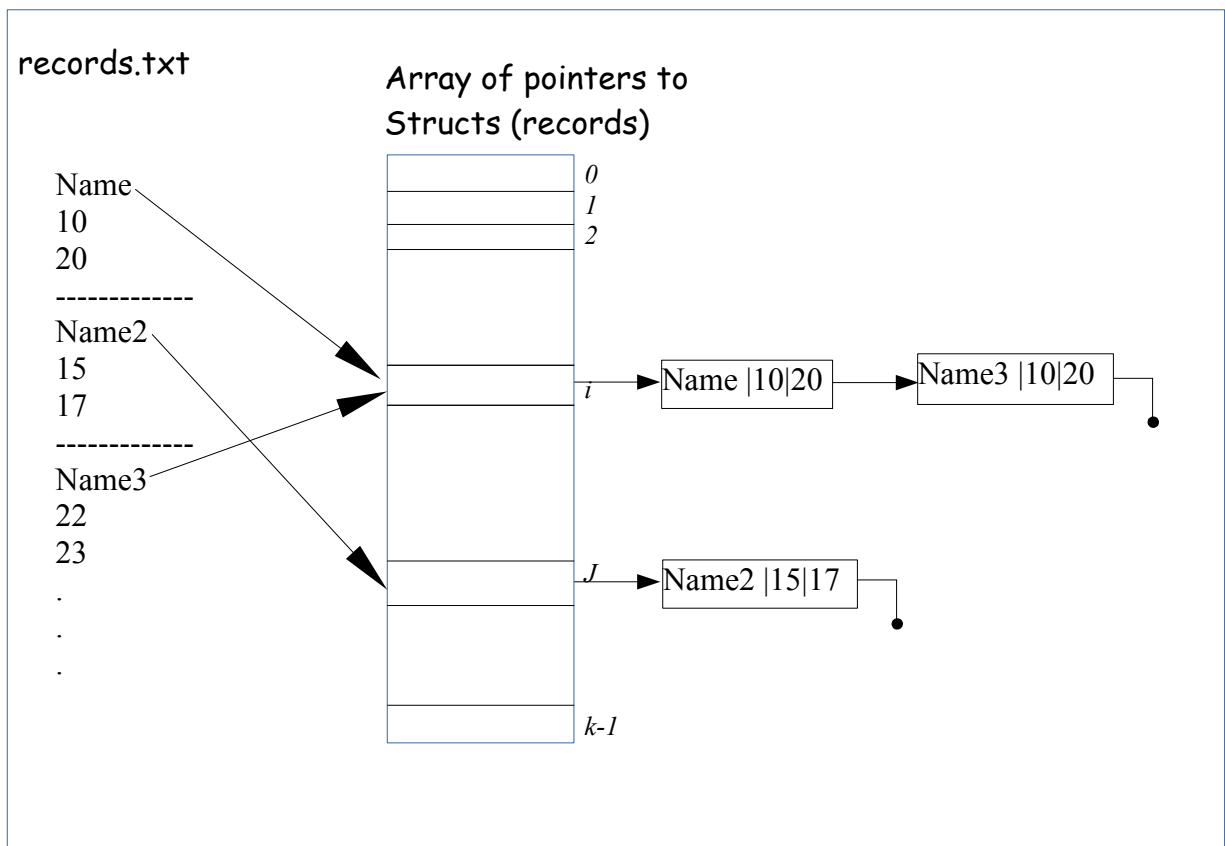
In this lab, you will expand on hashing (lab 4) in order to implement the linked list where the data are stored.

Introduction

The objective is to store the records of a file in a linked list such that it will be faster to search for any record.

The file is read record by record, each record is hashed to a specific location. If more than one record is mapped to the same location, it is stored in a linked list pointed to by the "location".

As shown in the following figure, the records are read from the file one by one, each record is hashed, and stored in the appropriate location in a linked list.



Specifications

Each record consists of a student name (last name, first, then optional middle name) followed by a new line.

Then two integers each one in a separate line, these two integers are the midterm mark and the final mark.

The records are **NOT separated by dotted line** the records are continuously stored each record in 3 lines. The dotted lines in the figure are for illustration purpose only.

The hashing function is the summation of all bytes in the name (including spaces, comma if any, but not the new line) modulo K.

the records are stored in a file named "records.txt"

Your program reads from the standard input the integer K

After reading K, the program opens the file records.txt and process all the records in the file. The result is number of linked lists (maximum K) each is pointed to by a pointer stored in array[hash(name)].

Then the program reads from the standard input a sequence of names (student names). For every name read, depends if the name is in the records.txt file or not.

- If the name is in the file, the program outputs three integers in one line
 - The two numbers in the record (midterm and final) and the number of search steps to locate the record. If the record is at the head of the list (like Name or name2 in the figure) the search steps is 1. If it is like Name3, the search steps is two, and so on.
 - For simple testing, use the format "%d %d %d\n" where there is one space between the %d's
- If the name is not in the file, the program outputs the following
 - "The name does not exist in the file" without the quotation marks **followed by a new line.**

Submit as submit 2031 L6 L6.c