Seoul National University

M1522.000900 Data Structure

Fall 2017, Kang

Homework 7: Searching (Chapter 9)

Due: November 30, 02:00 PM

Reminders

- The points of this homework add up to 100.
- Like all homeworks, this has to be done individually.
- Lead T.A.: Beunguk Ahn (beunguk.ahn@gmail.com)
- Please type your answers in English. Illegible handwriting may get no points, at the discretion of the graders.
- If you have a question about assignments, please upload your question in eTL.
- If you want to use slip days or consider late submission with penalties, please note that you are allowed one week to submit your assignment after the due date.

Remember that:

1. Whenever you are making an assumption, please state it clearly

-

Question 1

Jump search is a search algorithm which skips some of the items of an array to improve the performance. Let n be the number of items in the array. Then, the optimal size k of jump is $\sqrt{n}$. Answer the following questions [10 points]:

(1) What is the number of comparisons when we search the given sorted array for the element of 2^{74}?

$[2^1 \quad 2^2 \quad 2^3 \quad \ldots \quad 2^{100}]$ (a list of elements in power series)

$\sqrt{100} = 10, size\ of\ the\ step$

Stop jumping at 2^{80} -> 8 steps

1. Begins at 2^{70}, compare values forward until it reaches 2^{74} -> 4 steps = 8 + 4 = 12
2. Begins at 2^{80}, compare values backward until it reaches 27^4 -> 6 steps = 8 + 6 = 14
3. Other methods are also possible. If your method is acceptable (efficient enough), you will get full grade.

(2) What is the number of comparisons when we search the given sorted array for the element of 234?

[0 4 5 10 23 30 43 46 60 65 65 100 134 234 456 459]

$\sqrt{16} = 4, size\ of\ the\ step$

Stop jumping at 459 -> 4 steps

1. Begins at 100, compare values forward until it reaches 234 -> 2 steps = 4 + 2 = 6
2. Begins at 459, compare values backward until it reaches 234 -> 2 steps = 4 + 2 = 6
3. Other methods are also possible. If your method is acceptable (efficient enough), you will get full grade.

Question 2

Beunguk Ahn (marked as Ahn in the following contents), a TA at Data Structure, has to manage more than 130 students. Grading the students by eyes, searching through a spread sheet, is quite a work. Thus, he has decided to use searching methods discussed in the class.

Basically a student ID has the format of "XXXX-XXXXX" or "OO-OOOOO," but Ahn changes its format to "XXXXXXXXX", "19OOOOOOO". Students entered from 2000 year have the formal format, and students entered before 2000 have the latter format. For example, student IDs "2013-12345" and "97-54321" are converted into "201312345" and "199754321." We call this format *N_SID* which stands for "Numeric Student ID." Ahn will use an *N_SID* as a key for a hash function.

The minimum value of *N_SID* in the class is 199700000.
The maximum value of *N_SID* in the class is 201799999.

Answer the following questions to help Ahn.

Question 2-1

At first glance, Ahn decides to use a 10-slot **open hashing table** (the slots are numbered 0 through 9) for which he use the following function. The initial table is empty. [30 points]

$$h_1(N\_SID) = \lfloor N\_SID \ / \ 10000 \rfloor \ mod \ 10$$

(a) Show the final hash table that would result if you insert *N_SID*s as follows.

[201610005   201610001   199743532   201634532   201723453   200733222].

0	0	
1	201610005	→ 201610001
2	201723453	
3	201634532	→ 200733222
4	199743532	
5		
⋮		
9		

(b) How many collisions occur in the question (a) above? **2**

(c) What could be the problem of using this hash function h_1? Describe it in at most two sentences.

- It does not use all information from the data.
- The distribution of student is not uniform. Even though it is uniform, the distribution of hash values is not uniform.

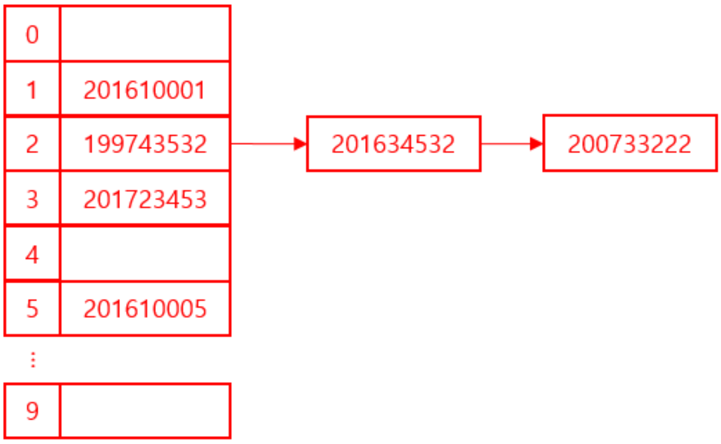
Question 2-2

Ahn decides to change the hash function as the following: [20 points]

$$h_2(N\_SID) = N\_SID \; mod \; 10$$

(a) Show the final hash table that would result if you insert *N_SID*s as follows.

[201610005 201610001 199743532 201634532 201723453 200733222].



(b) How many collisions occur in the question (a) above? **2**

Question 2-3

Suppose $p(n)$ is the probability of at least two of the n students sharing a same hash value. Assume N_SIDs are independently and uniformly distributed at the range of the minimum to the maximum. [20 points]

$p(n) = 1 \times (1 - \frac{1}{m}) \times (1 - \frac{2}{m}) \times \cdots \times (1 - \frac{\mathrm{n-1}}{m})$, where *m* is the size of the slot

(a) At which n, does $p(n)$ exceed 0.99 using the hash function $h_2(k)$ in the question 2-2?

When *m*=10, $p(n)$ exceeds 0.99 when *n*=9.

(b) At which n, does $p(n)$ exceed 0.99 using the hash function h_3?

$$h_3(N\_SID) = N\_SID \ mod \ 20$$

When *m*=20, $p(n)$ exceeds 0.99 when *n*=13.

Question 3

Consider inserting the keys 10, 21, 34, 45, 4, 88, 35, 91, and 59 into a hash table of length 11 using **open addressing**. (Slots are numbered 0 through 10). We will use the following function as a hash function. [10 points]

$$h(k) = k \ mod \ 11$$

(a) For linear probing, we use $p(k, i) = i$ for probing. Show the final hash table that would result if you insert the given keys.

Slot	0	1	2	3	4	5	6	7	8	9	10
Key	21	34	45	88	4	35	91	59			10

(b) For quadratic probing, we use $p(k, i) = i^2 + 3i$ for probing. Show the final hash table that would result if you insert the given keys.

Slot	0	1	2	3	4	5	6	7	8	9	10
Key	88	34	35	21	4	45		91	59		10

Question 4

Text compression is an example that uses Move-to-Front heuristic algorithm. For example, “the car on the left hit the car I left” could be compressed as “the car on 3 left hit 3 5 I 5”. [10 points]

Compress the following quote. Ignore the periods and commas.

> **I felt happy because I saw the others were happy and because I knew I should feel happy, but I was not really happy.**

I felt happy because 4 saw the other were 7 and 8 8 knew 2 should feel 7 but 5 was not really 6