

COMP20012 Tutorial 5: Hashing and Trees – The questions

1. Given a hash table size of 10 and hash function $h(x) = x \bmod 10$, show the result of inserting the data

4371, 1323, 6173, 4199, 4344, 9679, 1989

into

- a) an open hash table
 - b) a closed hash table using linear probing
 - c) a closed hash table using quadratic probing
 - d) a closed hash table using a secondary hash function $h_2(x) = 7 - (x \bmod 7)$
2. A room contains n people. Let $P(n)$ denote the probability that at least two people in the room share the same birthday. Denote by $Q(n)$ the probability that *no two people* in the room have the same birthday. Explain to your tutor why
- a) $P(n) = 1 - Q(n)$
 - b)

$$Q(1) = 1$$

$$Q(2) = Q(1) \frac{364}{365}$$

$$Q(3) = Q(2) \frac{363}{365}$$

$$Q(n) = Q(n-1) \frac{365 - n + 1}{365}$$

Write a short program to evaluate $P(n)$ and find the first value of n such that $P(n) > 0.5$, i.e. the smallest n such that, in a room containing n people there is a better than an even chance of two people sharing the same birthday.

What lesson should you draw from this regarding the likelihood of clashes occurring in hash tables?

3. Explain to your tutor what AVL trees are, and how their type invariant is maintained after insertion or deletion of elements. Show the result of inserting 2, 1, 4, 5, 9, 3, 6, 7 into an initially empty AVL tree.

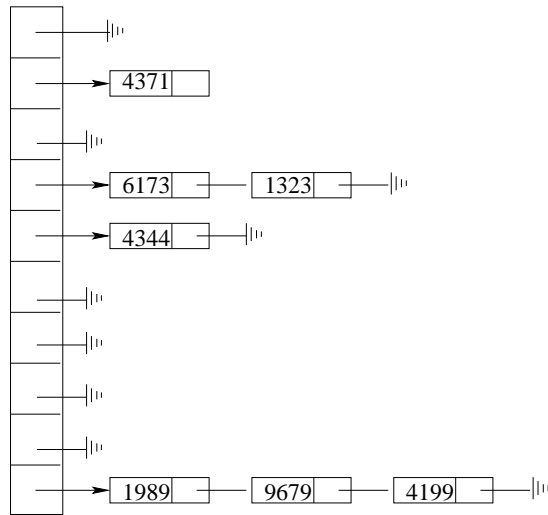
COMP20012 Tutorial 5: Hashing and Trees – Tutors notes

This tutorial may be a little on the light side, but if your group is anything like mine you probably have some catching up to do with the earlier tutorials.

Graham

Aim: This sheet is to help reinforce the material on hashing and tree balancing by asking students to run through some examples of these techniques. It also (in Question 2) shows how, by calculating probabilities, we may estimate the likelihood of collisions in hashing and therefore estimate the size of hash tables required.

1. a) Open hashing



b) Linear probing

=====									
0							9679	9679	
1		4371	4371	4371	4371	4371	4371	4371	
2								1989	
3			1323	1323	1323	1323	1323	1323	
4				6173	6173	6173	6173	6173	
5						4344	4344	4344	
6									
7									
8									
9					4199	4199	4199	4199	
=====									

c) Quadratic probing

=====									
0							9679	9679	
1		4371	4371	4371	4371	4371	4371	4371	
2									
3			1323	1323	1323	1323	1323	1323	
4				6173	6173	6173	6173	6173	
5						4344	4344	4344	
6									

```
7 ||      |      |      |      |      |      |
8 ||      |      |      |      |      |      |1989 |
9 ||      |      |      |4199 |4199 |4199 |4199 |
=====
```

d) Double hashing

```
=====
0 ||      |      |      |      |      |      |
1 ||4371 |4371 |4371 |4371 |4371 |4371 |
2 ||      |      |      |      |      |      |
3 ||      |1323 |1323 |1323 |1323 |1323 |
4 ||      |      |6173 |6173 |6173 |6173 |
5 ||      |      |      |      |      |9679 |
6 ||      |      |      |      |      |      |
7 ||      |      |      |      |4344 |4344 |
8 ||      |      |      |      |      |      |
9 ||      |      |      |4199 |4199 |4199 |
=====
```

The attempt to insert 1989 fails.

2. This question is about the (well-known?) von Mises Birthday ‘Paradox’. (Not really a paradox at all, just a surprising result)

$Q(n) = Q(n-1) \frac{365-n+1}{365}$, since, when the n th person enters the room, none of the previous $n-1$ must share a birthday and the n th person must have a birthday which misses the $n-1$ different birthdays of the people already there.

The smallest n is only 23, which is almost certainly counter-intuitive.

```
static void vonMises ()
{
    int n = 0;
    double q = 1;
    while (q > 0.5)
    {
        n++;
        q *= (365.0 - n + 1)/365.0;
        if (n % 5 == 0) {
            System.out.println(n + "\t" + (1-q));
        }
    }
    System.out.println("n is " + n + ", p is " + (1-q));
}
```

The moral of the tale is that that clashes in hash tables are far more likely than you probably expect.

3. Copies of slides on AVL trees can be found on the course web pages at <http://www.cs.man.ac.uk/~graham/COMP20012/lectures/graham/trees.pdf>.

The results of inserting into the tree are as follows. I’ve missed the first three out.

