Algorithms and Data Structures

Arnav Singh

April 2020

10 Sheet 10

10.1 Hash Tables

(a) Double Hashing

Here are given the sequence $< 3, 10, 2, 4 >$ with
$h_1(k) = k \bmod 5$ and $h_2(k) = 7k \bmod 8$ and hashtable of size 5
So firstly we have the hash table as given below. We are aksed to insert 3 first so according to the formula.

$h_1(3) = 3 \bmod 5 = 3$ and this is with no collision since the hash table the cell at index 3 is empty. Thus we get:

0	1	2	3	4
			3	

Next we insert 10
$h_1(10) = 10 \bmod 5 = 0$ and this is with no collision since the cell at index 0 is empty. Thus we get:

0	1	2	3	4
10			3	

Next we insert 2
$h_1(2) = 2 \bmod 5 = 2$ and this is with no collision since the cell at index 2 is empty. Thus we get:

0	1	2	3	4
10		2	3	

Lastly we insert 4
$h_1(4) = 4 \bmod 5 = 4$ and this is with no collision since the cell at index 4 is empty. Thus we get:

0	1	2	3	4
10		2	3	4

Thus overall no collision occurred in the hashing process and cell 1 is still empty.

(b) Implemented Hash table check zip file

10.2 Greedy Algorithms

(a) Activity Selection

Lets consider a counter example. Suppose we have the following activities. [(3,7),(6,9)(8,14)] If we implement the activity selection using greedy approach with the shortest duration, it will select (6,9) since it has the shortest duration of 3. Next it will try to select (3,7) since it has the second shortest duration. It can't select (3,7) because it overlaps. Next it will try to select (8,14) but this is also not possible since the activity overlaps. Thus we get (6,9) as the greedy solution.
However, if we consider the globally optimum solution it would contain both (3,7) and (8,14). Hence in this case it will not always produce the globally optimum solution.