Algorithms and Data Structures Homework 11

Matheas-Roland Borsos

May 1, 2023

1 Hash Tables

1.1 Given the sequence 3, 10, 2, 4, apply the double-hashing strategy for open addressing to store the sequence in the given order in a hash table of size m = 5 with hash functions h1(k) = k mod 5 and h2(k) = 7k mod 8. Document all collisions and how they are resolved. Write down your computations:

$$h(k, i) = (h1(k) + i * h2(k)) \text{ mod m}$$
$$h(k, i) = (k \text{ mod } 5 + i * (7k \text{ mod } 8)) \text{ mod m}$$

For 3: h(3,1) = (3 mod 5 + 1 * (7*3 mod 8)) mod 5 = (3 + 5) mod 5 = 3

For 10: h(10,1) = (10 mod 5 + 1 * (7*10 mod 8)) mod 5 = (0 + 6) mod 5 = 1

For 2: h(2,1) = (2 mod 5 + 1 * (7*2 mod 8)) mod 5 = (2 + 6) mod 5 = 3, but position 3 is occupied so we calculate h(2,2) = (2 mod 5 + 2 * (7 * 2 mod 8)) mod 5 = (2 + 12) mod 5 = 4

For 4: h(4,1) = (4 mod 5 + 1 * (7*4 mod 8)) mod 5 = (4 + 4) mod 5 = 3, but position 3 is occupied so we calculate h(4,2) = (4 mod 5 + 2 * (7 * 4 mod 8)) mod 5 = (4 + 8) mod 5 = 2

1.2 Implement a hash table that supports insertion and querying with open addressing using linear probing. Select an h' function and explain why your selected h' is well suited for your test data.

My h' function works using the division method: key mod m, where m is the size of the Hash Table. The chosen m is 25, because it is a number not too close to the powers of 2 and the type of keys we are storing are integers (in this case unique).

2 Greedy Algorithms

2.1 Show that a greedy algorithm for the activity-selection problem that makes the greedy choice of selecting the activity with shortest duration may fail at producing a globally optimal solution.

In order to show that the greedy choice of this algorithm may fail, an example of failure must be provided. The goal of this algorithm is to choose intervals in such a way that it maximizes the number of intervals chosen, while they are non-conflicting or compatible with each other. In the following example, the greedy choice of selecting the activity with the shortest duration fails at producing a globally optimal solution.

Example: The interval I is: (2, 7),(5, 10),(7, 12),(4,8). The solution that the greedy choice of the shortest duration gives in is (4,8)

The algorithm arrives at this solution as the interval with the shortest duration is interval (4, 8) and no other interval offers the possibility to be added to the solution set. The optimal solution to the goal of the algorithm is the set of intervals (2, 7),(7, 12). So it fails.

2.2 Assuming an unsorted sequence of activities, derive a greedy algorithm for the activity-selection problem that selects the activity with the latest starting time. Your solution should not simply sort the activities and then select the activity.

The algorithm of finding the optimal solution to the activity selection maximization problem is implemented based on a iterative approach of the greedy activity. This algorithm doesn't need to prepossess the array of activities by sorting them.

This algorithm sets up a loop which will iterate until the given array of activities is empty. It will choose in a iterative mode the latest starting time activity of the current activities in the array, remove it from the array and it will decide to add the activity onto the optimal solution set array depending if the activity is in conflict with the previously added activity of the optimal solution set array.

The time complexity of my greedy algorithm is O(n^2), because it requires 2 nested loops inside of each other to create the optimal solution set array, performing worse than the greedy approach given in the book and slides which is O(nlog(n)) due to prepossessing the array (time complexity of O(n) omitting sorting).