David Lee
dyl30

Part 1 (15 Points)

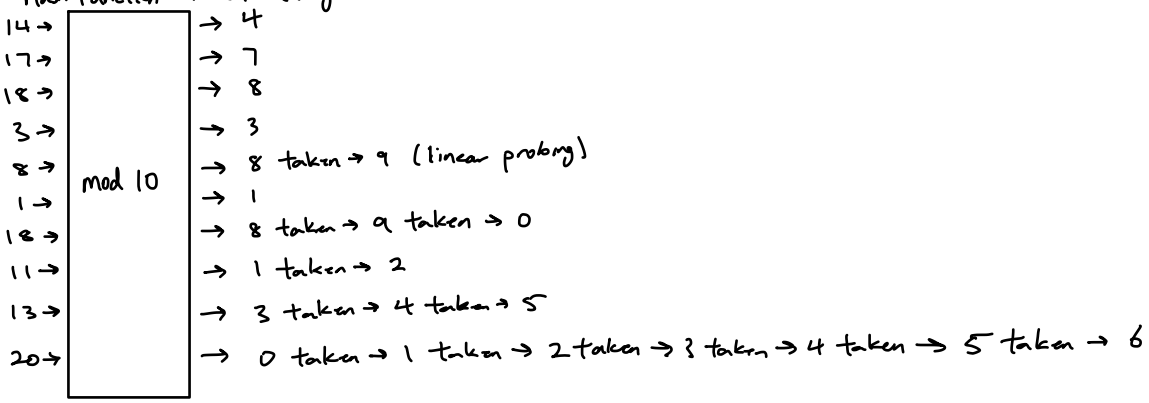
We can use linear probing to solve this problem when building a hash table. Match the following numbers to the hash table with length 10. Please fill in the final result in the blank table. (Partial scores only apply to cells with correct answers)

14	17	18	3	8	1	18	11	13	20

Your result:

0	1	2	3	4	5	6	7	8	9
18	1	11	3	14	13	20	17	18	8

Hash Function : mod 10 (length 10)



Part 2 (10 Points, 30 Points)

I have an array [2,12,22,32,42,52,62,72,82,92].

a) Could I still use linear probing to build the hash table? Why? Is there any other method to reduce the disadvantage?

Yes we can still use linear probing to build a hash table. It is because linear probing continues until it finds an empty slot. A disadvantage is primary clustering. All the numbers end in 2 so when applying a simple modulo hash function, all of them would hash to the same initial index. To reduce disadvantage, we can use a better hash function, use quadratic or double probing or increase table size.

b) Now the hash table is double-sized. We also add numbers in Part 1 to this hash table. To double hash the table, we still use $hash(key) = key \bmod table\_size$ as the first step but bring a new second function $hash(key) = 7 - (key \bmod 7)$. Please show me the final answer.

8	17	2	62	82	20	11	18	22	42	18	3	12	1	14	32	52	72	92	13

Array 1

hash1(key) = key mod 20
hash2(key) = 7 - (key mod 7)

k	hash1	hash2
14	14	7
17	17	4
18	18	3
3	3	4
8	8	6
1	1	6
18	18	3
11	11	3
13	13	1
20	0	1

Array 2b

hash1(key) = key mod 20
hash2(key) = 7 - (key mod 7)

k	hash1	hash2
2	2	5
12	12	2
22	2	6
32	12	3
42	2	7
52	12	4
62	2	1
72	12	5
82	2	2
92	12	6

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
8	17	2	62	82	20	11	18	22	42	18	3	12	1	14	32	52	72	92	13