CP2410 Practical 08

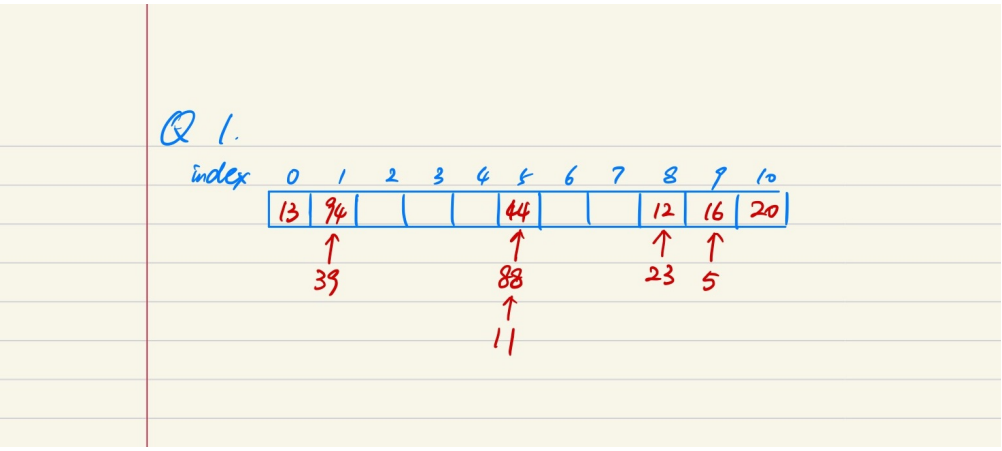
Sihan Chen, jcu ID: 14187662

Question 1

Hash function: $(3i+5) % 11$

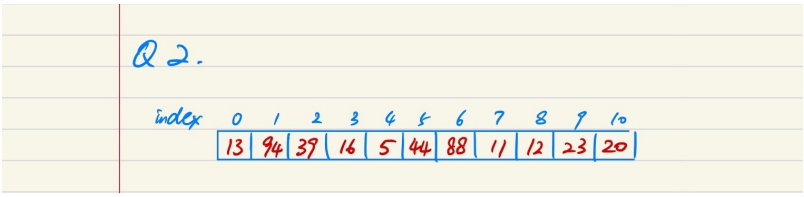
Key	Hash Value
12	8
44	5
13	0
88	5
23	8
94	1
11	5
39	1
20	10
16	9
5	9

and handling collisions with chaining will give us a hash table below:



Question 2

If the collisions are handled by linear probing, we will have a hash table look like this:



Question 3

If the collisions are handled by quadratic probing

$A[h(i) + j^2]\ (mod\ N), for\ j = 0, 1, \cdots, N - 1$

we can build a simple python function helps us to check quadratic probing new values

```
>>> def qh(n):
...     for i in range(11):
...             print(i, ((3*n)+5+i^2)%11)
...
```

and insert elements up to the point of failure, we will have a hash table look like this:

Q3.

index	0	1	2	3	4	5	6	7	8	9	10
	13	94	39	11		44	88	16	12	23	20

88 collides at 5, j h(i)
1 6

23 collides at 8, j h(i)
1 9

11 collides at 5, j h(i)
1 6
2 9
3 3

39 collides at 1, j h(i)
1 2

16 collides at 9. j h(i)
1 10
2 2
3 1

5 collides at 9. j h(i)
1 10
2 2
3 7
4 3
5 1
6 1
7 3
8 7
9 2
10 10

Since we cannot get a hash value of 4, the hash table fails to insert 5

Question 4

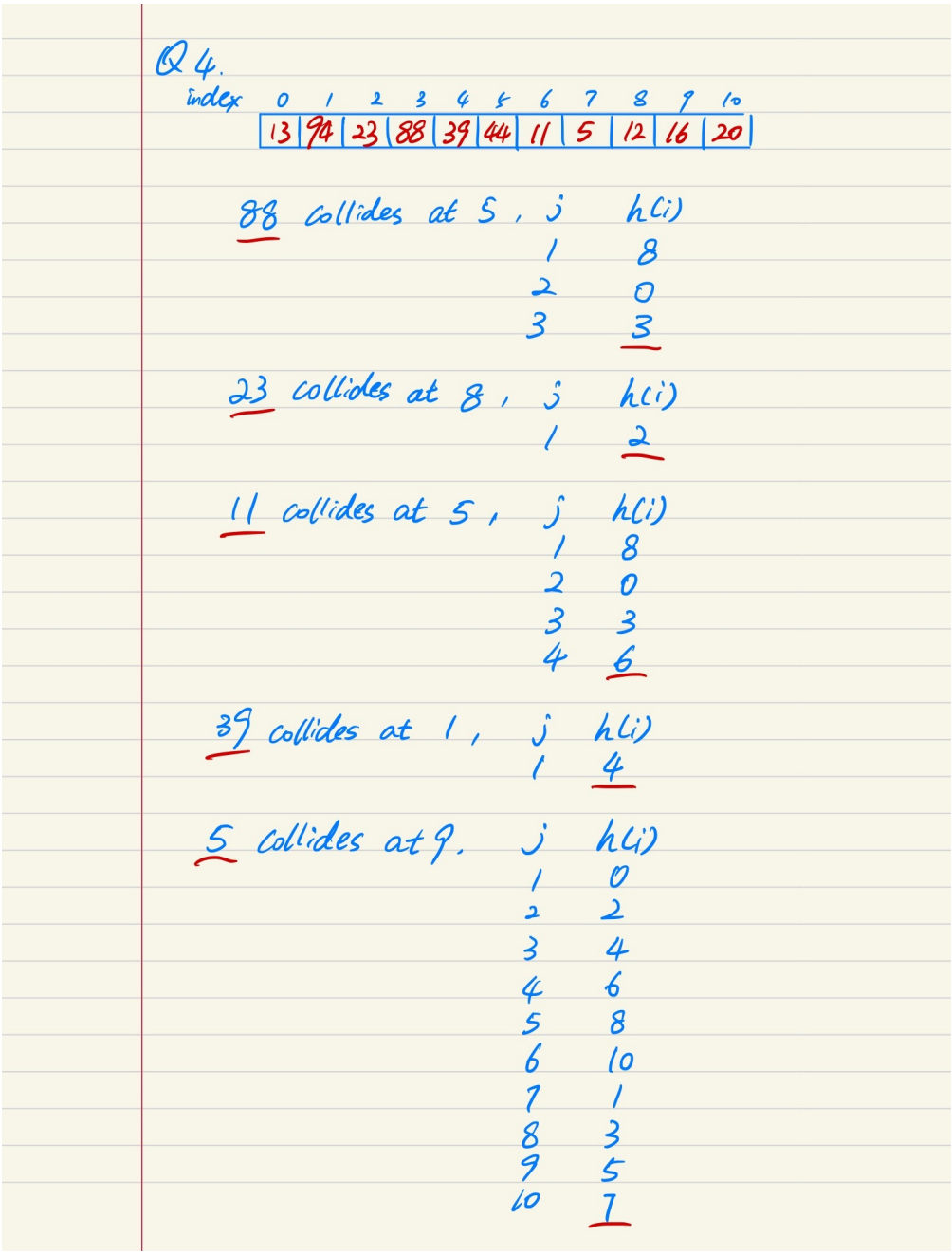
If the collisions are handled by double hashing

$H(i) = (h(i) + j*h'(i))\ mod\ N, for\ j = 0, 1, \cdots, N - 1$

we can build a simple python function helps us to check quadratic probing new values

```
>>> def sec_h(n):
...     return 7 - (n%7)
...
>>> def h(n):
...     return (3*n + 5) % 11
...
>>> def dh(n):
...     for i in range(11):
...             print(i, (h(n) + i * sec_h(n))%11)
```

the final hash table would look like this:



Question 5

Question 6