



AD2\_Assig...

AD2\_Assignment03\_AI.pdf

Donnerstag, 25. November 2021 10:05

## Assignment 03

### 2. Double Hashing

**12 points**

Apply the **double hashing** algorithm presented in the exercise using **pen & paper**.

In contrast to example 1, this algorithm doesn't need additional memory for the insertion of colliding elements. In case of collisions no overflow list is used, instead the elements are stored at other vacant positions in the hash table using a 2<sup>nd</sup> hash function. The two hash functions  $h1$  and  $h2$  are defined as

$$h1(k) = k \bmod N$$

$$h2(k) = 1 + k \bmod (N-1)$$

where  $k$  is the key and  $N$  is the length of the hash table.

Resolve any collisions by (repeatedly) applying the following probing sequence:

$$h1 = (h1 + h2) \% N$$

#### a) Prefilling of table

Fill the following table using the provided insert statements:

$$N = 13$$

For each digit also provide the probing sequence (indices separated by commas).

*insert(19)*

0	1	2	3	4	5	6	7	8	9	10	11	12
						19						

Probing sequence: (6)

↳ "( )" olenotes first insertion attempt, i.e. where  $h_2$  was never used.

*insert(6)*

0	1	2	3	4	5	6	7	8	9	10	11	12
6						19						

Probing sequence: (6), 0

**Assignment 03***insert(11)*

0	1	2	3	4	5	6	7	8	9	10	11	12
6						19					11	

Probing sequence: (11)

*insert(26)*

0	1	2	3	4	5	6	7	8	9	10	11	12
6			26			19					11	

Probing sequence: (0), 3

*insert(14)*

0	1	2	3	4	5	6	7	8	9	10	11	12
6	14		26			19					11	

Probing sequence: (1)

### Assignment 03

#### b) Insert of student ID

Fill the table, that is now prefilled with keys from part a), with the three rightmost digits of your student id. Again, note the probing sequence.  
Please make a note in case one of the digits of your student id can not be entered.

e.g.: student id = k12345**678** -> *insert(8)*, *insert(7)*, *insert(6)*

k 12 005 **[6 0 4]**

Student id: k x x x x x **6 0 4**  
d6 d7 d8

$\leq 4$   
*insert(d8)*

0	1	2	3	4	5	6	7	8	9	10	11	12
6	14		26	4		19					11	

Probing sequence: (4)

$= 0$   
*insert(d7)*

0	1	2	3	4	5	6	7	8	9	10	11	12
6	14	0	26	4		19					11	

Probing sequence: (0), 7, 2

**Assignment 03***insert(d6)*

0	1	2	3	4	5	6	7	8	9	10	11	12
6	14	0	26	4		19				11		

Probing sequence: (6), 0 → algorithm finds: key 6 already inserted!

! cannot be entered!

(would e.g. return False)

{ if for some reason duplicate keys result be allowed: probing sequence = (6), 0, 7 ↴

**Submission**

Put your source files and a pdf for part 2 into a ZIP archive and name the file **k12345678-assignment03.zip** where k12345678 should reflect your student ID.

insert  
duplicate  
key at  
index 7