

Today's announcements:

MP6 available, due , 11:59p. EC due , 11:59p

Hash functions:

- Computed in $O(1)$ time.
- Deterministic.
- Satisfy the SUHA.

Hashing: summary and the plan for the day...

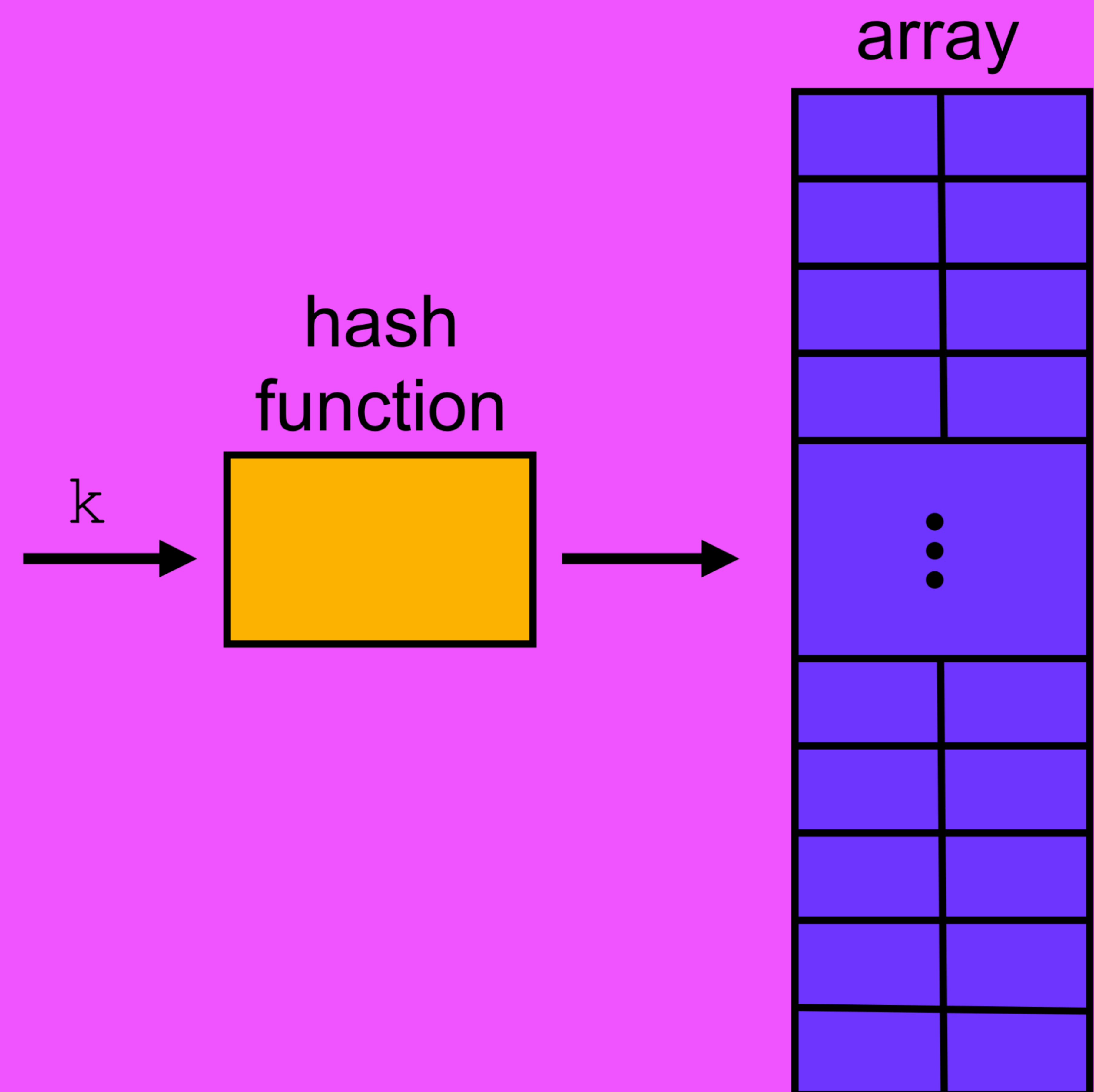
client code

declares an object
of ADT dictionary

```
dict<ktype, vtype> d;
```

ex: insert is `d[k] = v;`

class dict



Hash functions:

- Computed in $O(1)$ time.
- Deterministic.
- Satisfy the SUHA.

collision resolution strategy:

Let's hash short strings (8 characters)

```
int StringHashFunc(string str, int TABLE_SIZE)
{
    int sum = 0;

    for(int i = 0; i < str.size(); i++)
        sum += str[i]

    return sum % TABLE_SIZE;           // Compression
}
```

Decimal	Character
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j

A better hash function for short 8-character strings

a b a b a b a b

a ' s D - t b % r

Binary	Decimal	Character
01 1 00001	97	a
01 1 00010	98	b
01 1 00011	99	c
01 1 00100	100	d
01 1 00101	101	e
01 1 00110	102	f
01 1 00111	103	g
01 1 01000	104	h
01 1 01001	105	i
01 1 01010	106	j

Needless to say, a function that is exceptionally good for a specific kind of data may have dismal performance on data with different distribution.

An example of hashing longer strings

Given: 8 character strings are easy to hash

The idea: Select 8 random positions from long strings and hash that substring.

A bunch of strings:

Lookyhere, Huck, being rich ain't going
No! Oh, good-licks; are you in real dead
Just as dead earnest as I'm sitting here
nto the gang if you ain't respectable, y
Can't let me in, Tom? Didn't you let me
Yes, but that's different. A robber is m
irate is -- as a general thing. In most
Now, Tom, hain't you always ben friendly
ut, would you, Tom? You wouldn't do that
Huck, I wouldn't want to, and I DON'T wa
ay? Why, they'd say, 'Mph! Tom Sawyer's
t!' They'd mean you, Huck. You wouldn't
uck was silent for some time, engaged in
Well, I'll go back to the widder for a m
can come to stand it, if you'll let me
All right, Huck, it's a whiz! Come along
Will you, Tom -- now will you? That's go

An example of hashing longer strings

Given: 8 character strings are easy to hash

The idea: Select 8 random positions from long strings and hash that substring.

A bunch of strings:

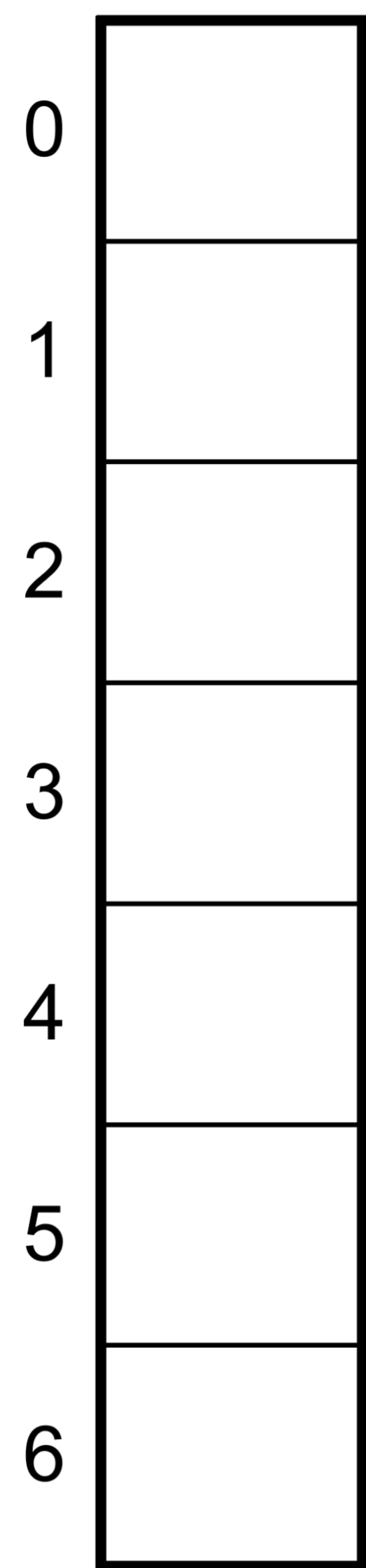
```
http://en.wikipedia.org/wiki/Le%C5%9Bna_Grobla
http://en.wikipedia.org/wiki/Blow_the_Man_Down
http://en.wikipedia.org/wiki/Swen_K%C3%B6nig
http://en.wikipedia.org/wiki/2/7th_Cavalry_Commando_Regiment_(Australia)
http://en.wikipedia.org/wiki/Salman_Ebrahim_Mohamed_Ali_Al_Khalifa
http://en.wikipedia.org/wiki/Alice_High_School
http://en.wikipedia.org/wiki/Beautiful,_Dirty,_Rich
http://en.wikipedia.org/wiki/RFA_Sir_Bedivere_(L3004)
http://en.wikipedia.org/wiki/Birthright_(band)
http://en.wikipedia.org/wiki/Jacky_Vimond
http://en.wikipedia.org/wiki/Vachon
http://en.wikipedia.org/wiki/McCarthy_%26_Stone
http://en.wikipedia.org/wiki/Salisbury,_New_Hampshire
http://en.wikipedia.org/wiki/A_Line_of_Deathless_Kings
http://en.wikipedia.org/wiki/Newfoundland_Irish
http://en.wikipedia.org/wiki/Beatrice_Politi
http://en.wikipedia.org/wiki/Bona_Sijabat
```

Collision handling - Separate Chaining: (an example of open hashing)

$S = \{16, 8, 4, 13, 29, 11, 22\}$

$|S| = n$

$h(k) = k \% 7$



	Worst case	Under SUHA
Insert		
Remove/find		

Collision Handling - Probe based hashing: (example of closed hashing)

$S = \{16, 8, 4, 13, 29, 11, 22\}$

$|S| = n$ $h(k) = k \% 7$

0	
1	
2	
3	
4	
5	
6	

Try $h(k) = (k + 0) \% 7$. If full...

try $h(k) = (k + 1) \% 7$. If full...

try $h(k) = (k + 2) \% 7$. If full...

try...

Probe based hashing – 2 problems...

Removals:

0	22	
1	8	
2	16	
3	29	
4	4	
5	11	
6	13	

Clustering:

