Msc in Informatics & Multimedia
Department of Informatics Engineering TEI of Crete

Bloom Filters

An Interactive scriptable implementation

Τσαγκατάκης Γιάννης
jtsagata@gmail.com

What is a bloom filter

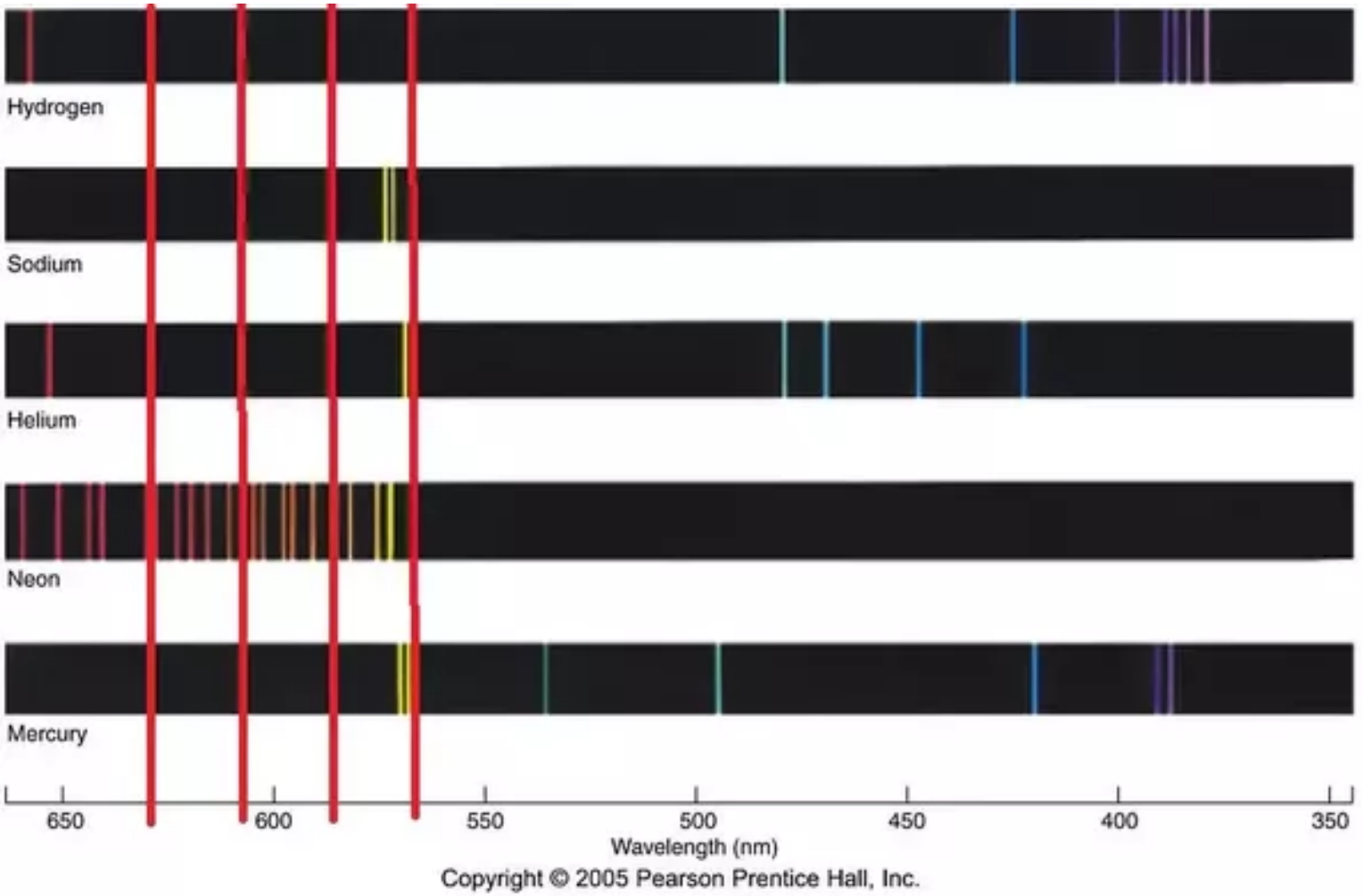
- **A set like data structure**
  - a bit array of m bits
  - k different hash functions
    - independent and uniformly distributed
    - md5, hashmix, crypto hashes, murmur, fnv
- **Operations**
  - `add(data)`
  - `exists?(data)`
- **Missing operations !!**
  - `delete, count, enumerate`
- **Operation `exists?()` is not working !**

It’s a useless data structure

- Spell Checkers
- Forbidden password lists
- Bad sites list (chrome)
- Yahoo mail (contact list)
- Internet cache protocol (ICP) Request handling
- Bitcoin network (SPV nodes)
- medium ,avoid recommend previously read articles
- Squid web proxy, for cache digests
- Databases: postgresq, cassandra, HBase, google Bigtable

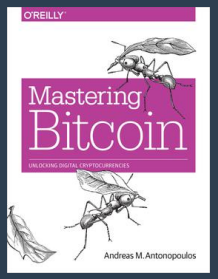
Physical analogy

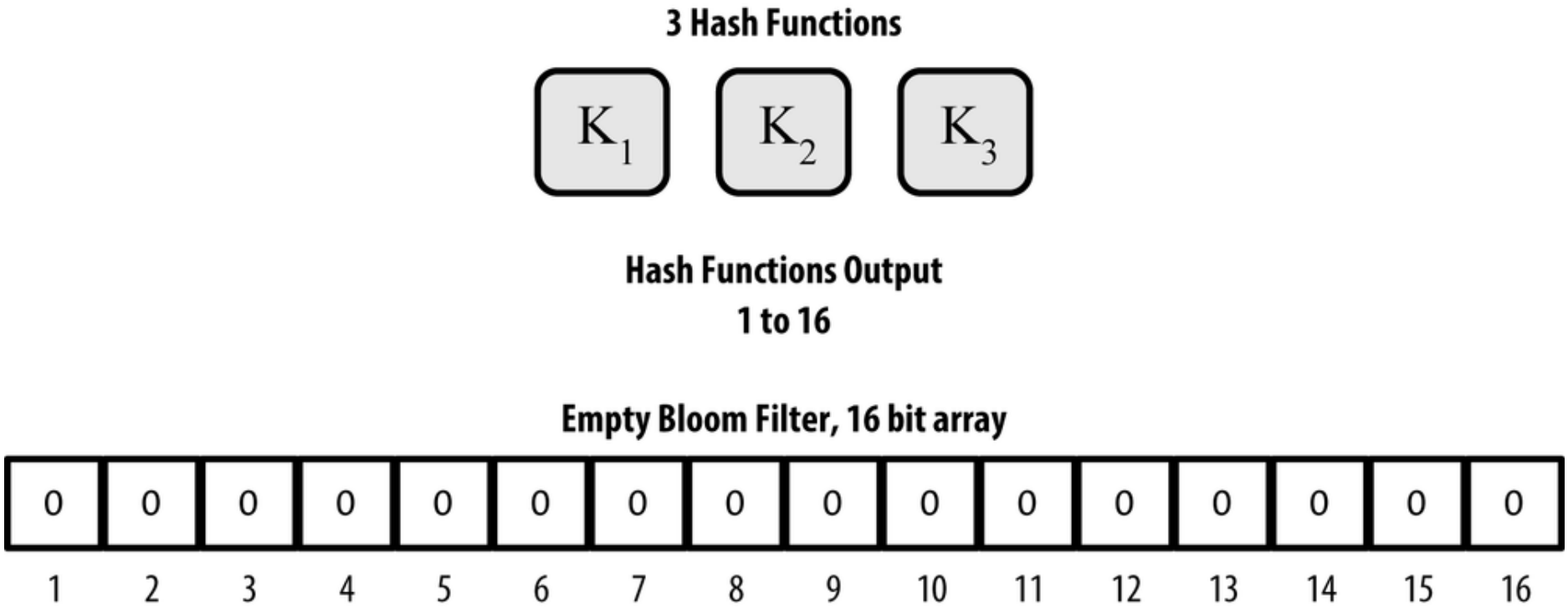
Is it possible that my set contains a particular element ?



Quora: How do bloom filters work?

The data structure : Organization

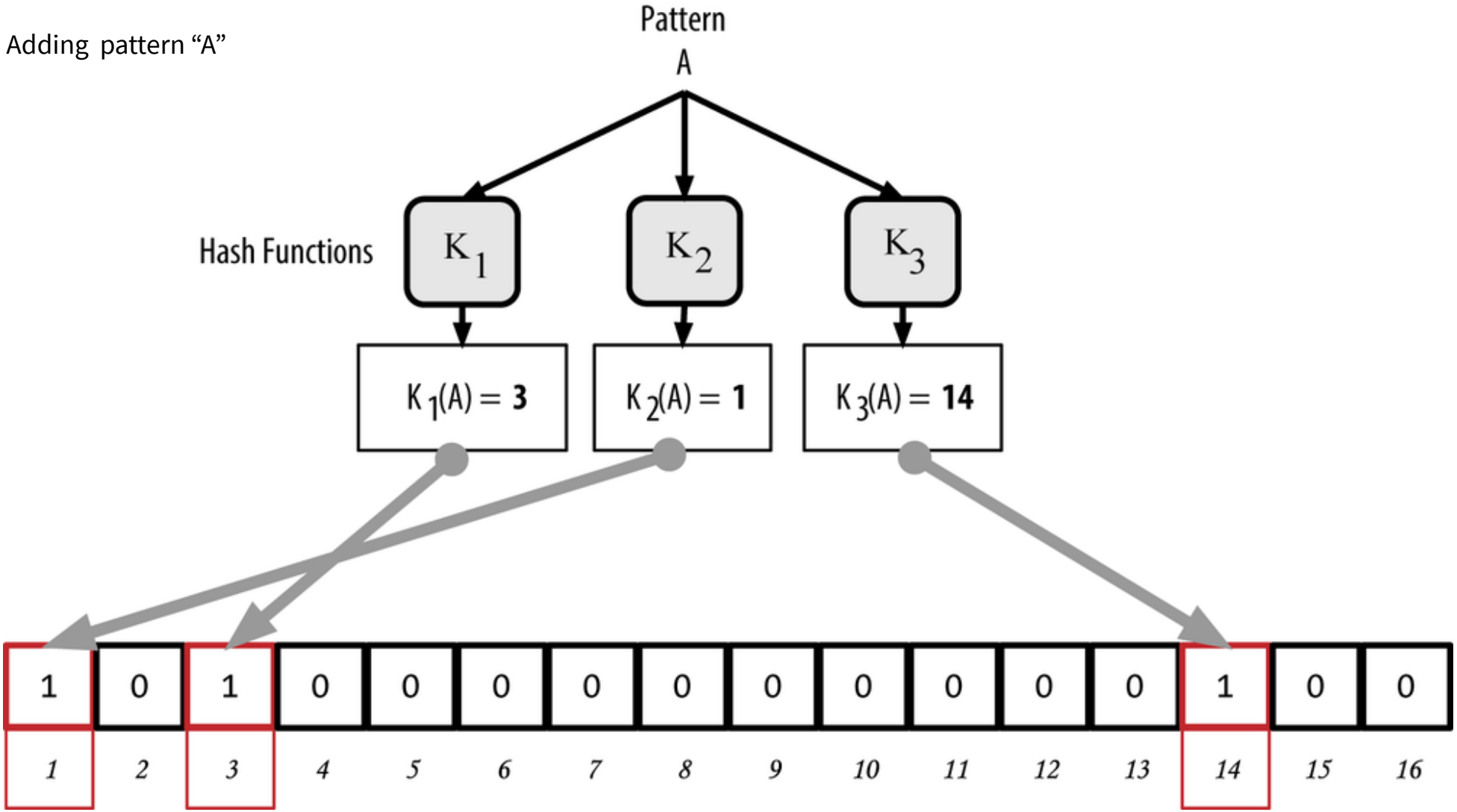




An example of a bloom filter, with a 16-bit field and three hash functions.

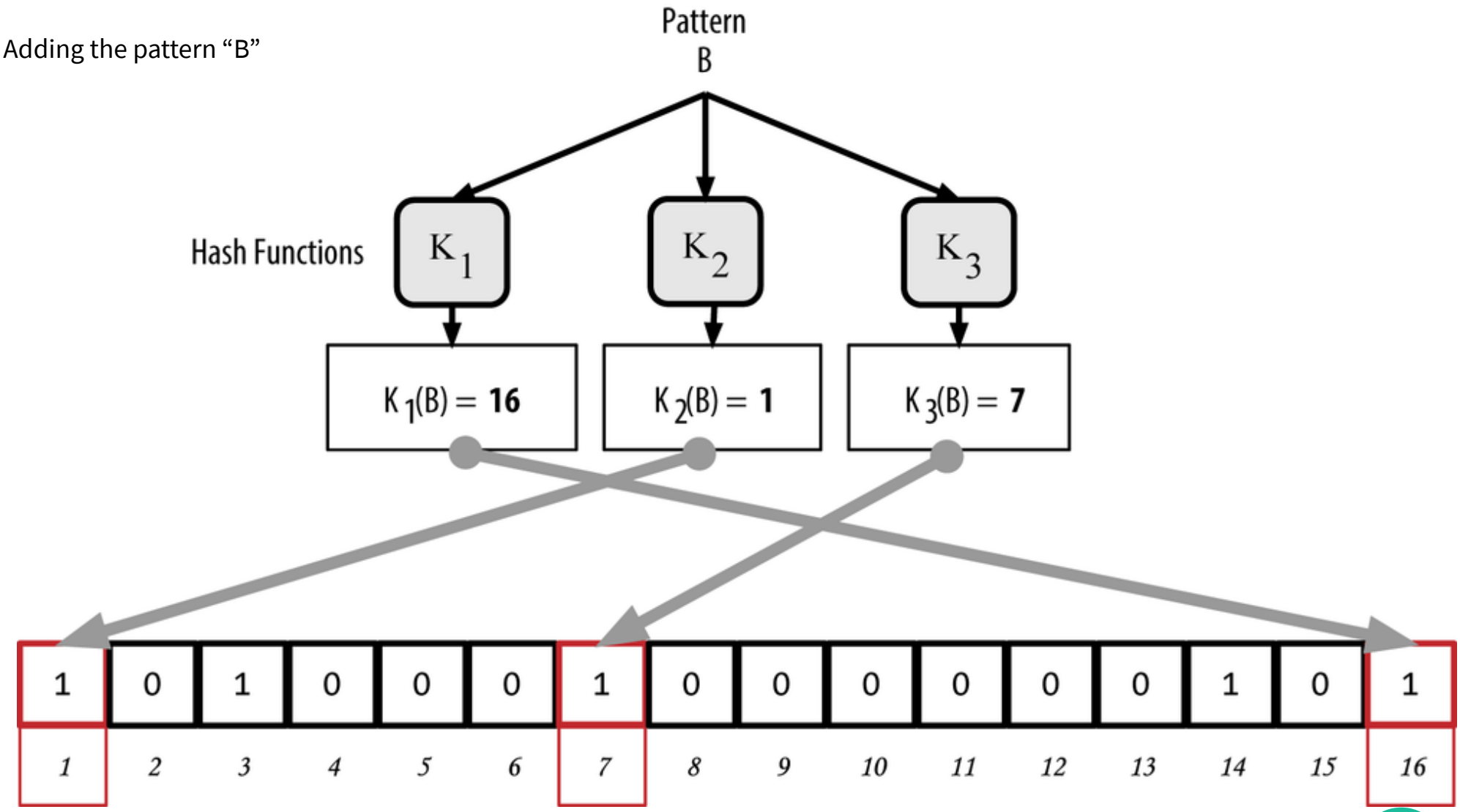
The data structure : Inserting

Adding pattern “A”

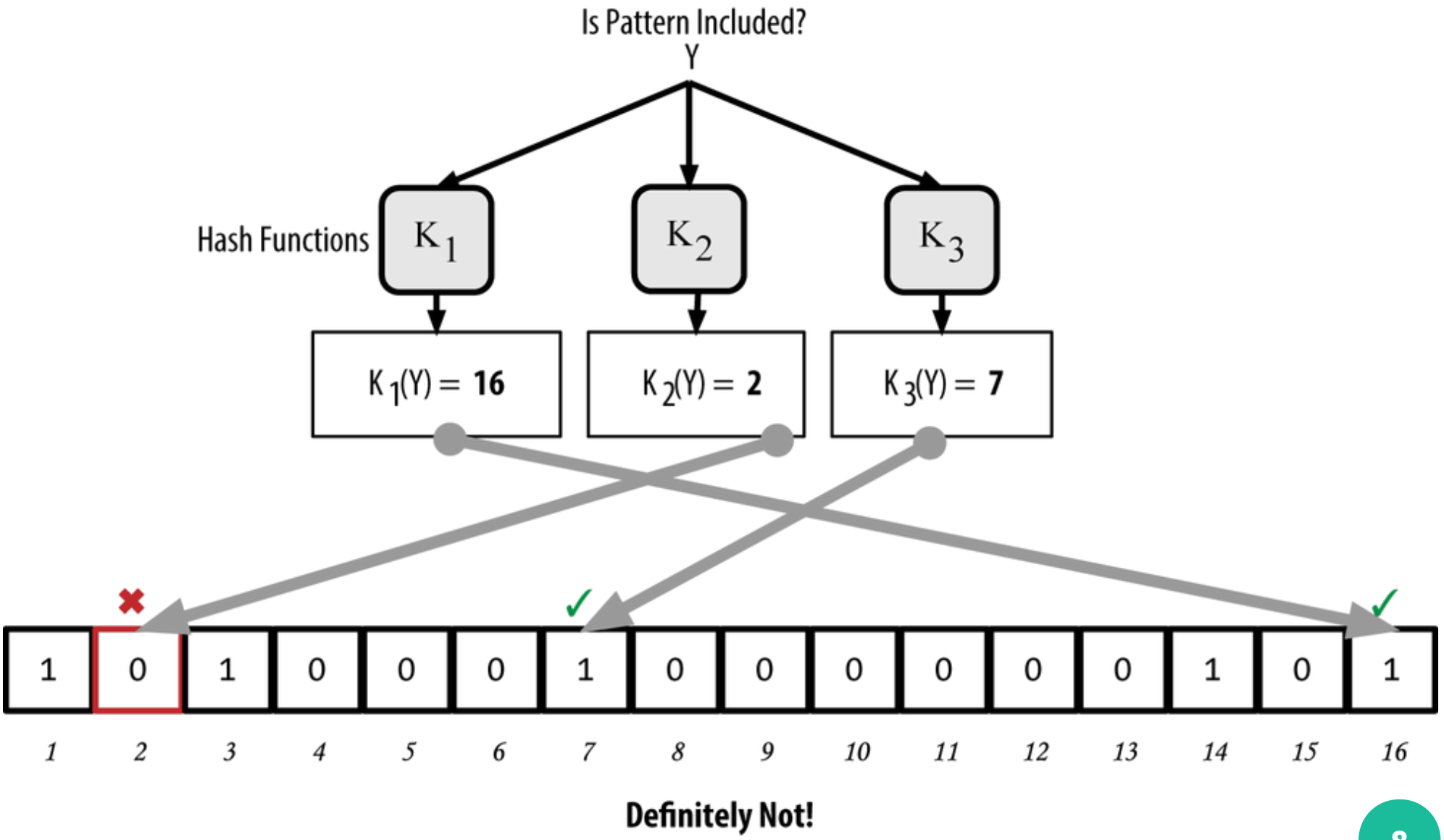


The data structure : Inserting more

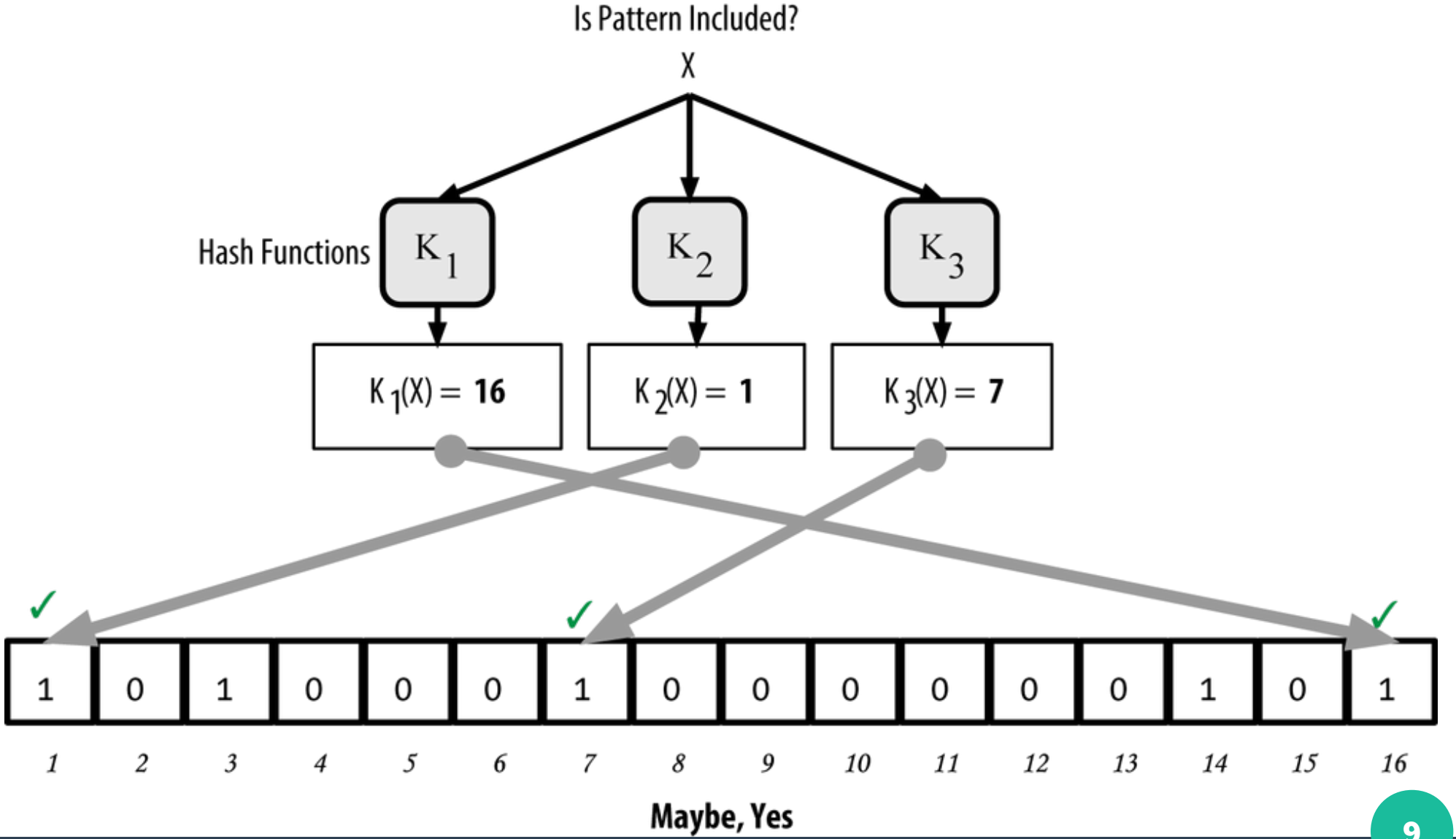
Adding the pattern “B”



The data structure : Testing



The data structure : Testing



The Verdict

	True	False
Exists	TRUE POSITIVE	FALSE NEGATIVE Never !
Absent	FALSE POSITIVE $\left(1 - e^{-kn/m}\right)^k$	TRUE NEGATIVE

Probably YES /Always NO

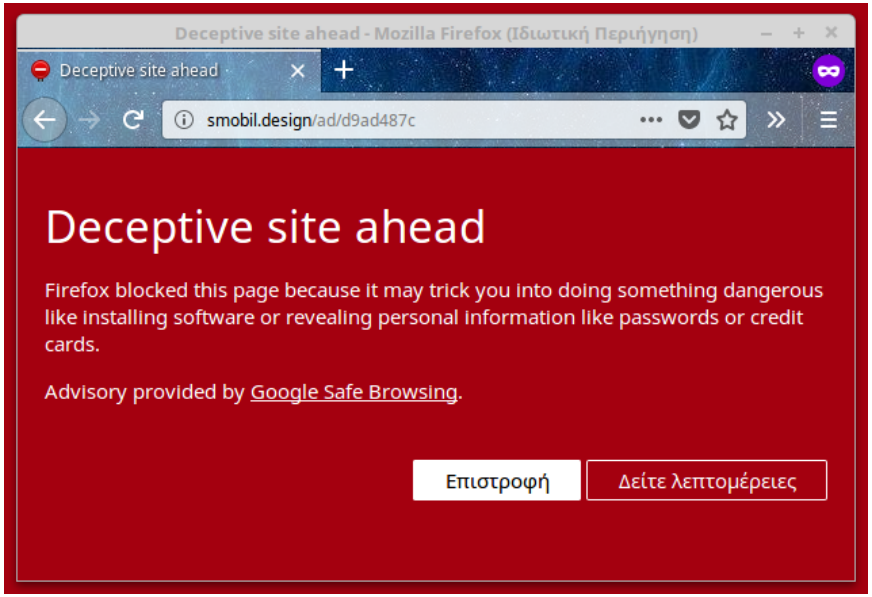
$$n^* = -\frac{m}{k} \ln\left[1 - \frac{X}{m}\right]$$

$$k = -\frac{\ln p}{\ln 2} = -\log_2 p.$$

$$m = -\frac{n \ln p}{(\ln 2)^2}$$

Example: Chrome Safe Browsing

- **The problem**
  - 1 million URLs of malicious web sites
  - Is this URL safe to browse ?
  - Avoid going at the network.
- **The solution**
  - A 18MB bloom filter
  - 9.6 bits per site (almost a byte)
  - False positive 1%
    - Go and check at a web service



http://blog.alexyakunin.com/2010/03/nice-bloom-filter-application.html

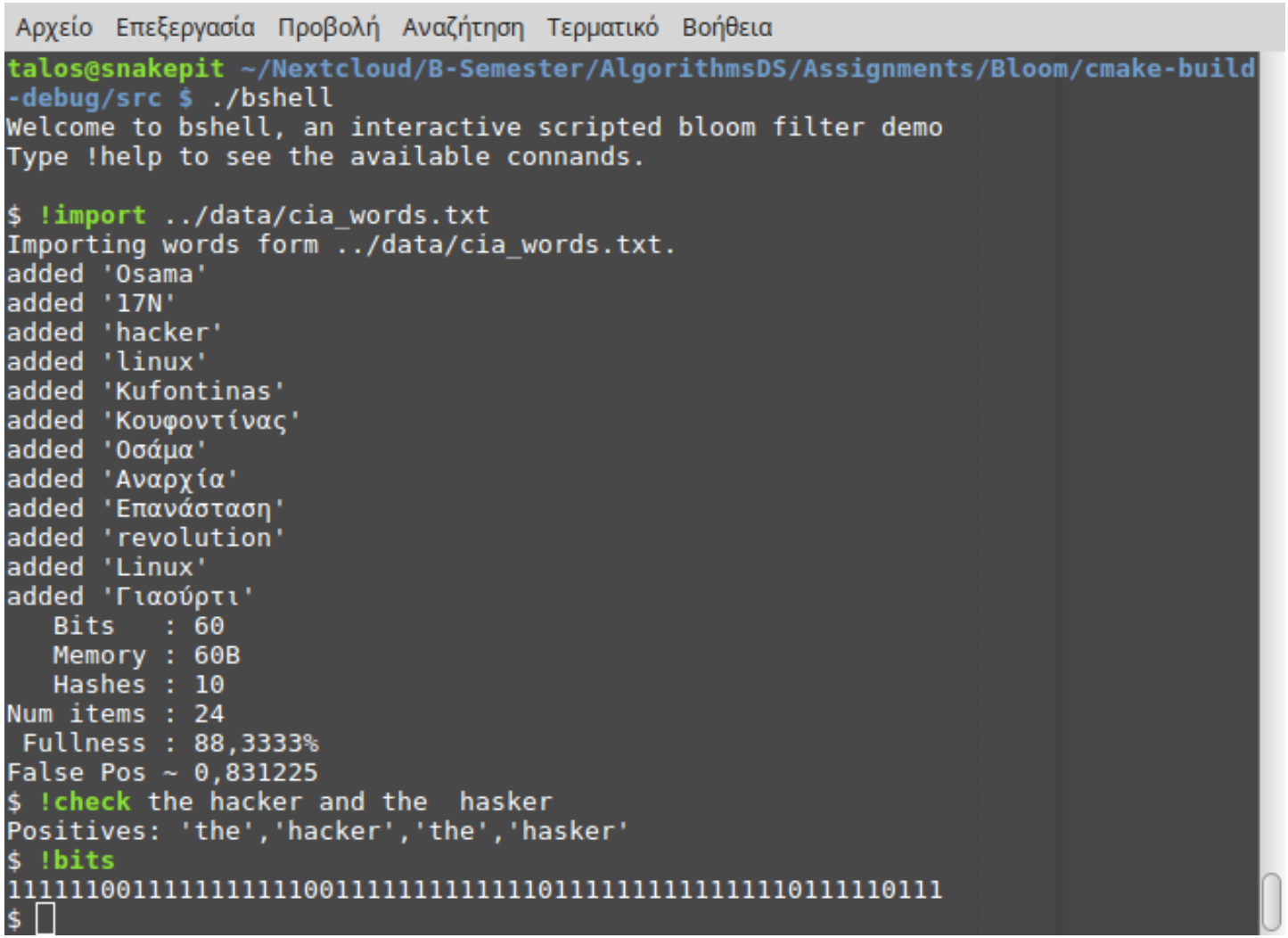
Tools used

- **C++14**
  - Clang++ v7
  - Boost libraries
  - AmokHuginnsson/replxx
- **Clion IDE**
- **Cmake / ninja**
- **Doxygen with LaTeX**

Extensions and alternatives

	Cuckoo Filter	Standard Bloom Filter	Counting Bloom Filter
Insert	Variable. O(1) amortized longer as load factor approaches capacity	Fixed. O(k)	Fixed. O(k)
As load increases	FPP trends toward desired max Insertions *may* be rejected if counting or deletion support is enabled	FPP trends toward 100% Insertions cannot be rejected	FPP trends toward 100% Insertions *may* be rejected
Lookup	O(1) Maximum of two buckets to check	O(k)	O(k)
Count	O(1) minimal suport: max == entries per bucket X 2	*unsupported*	O(k)
Delete	O(1) Maximum of two buckets to inspect	*unsupported*	O(k)
Bits per entry	smaller when desired FPP <= 3%	smaller when desired FPP > 3%	larger than Cuckoo & Standard Bloom multiplied by number of bits per counter
Bits per entry	$1.05\ [\ \log_2(1/FPP) + \log_2(2b)\ ]$ best when FPP <= 0.5% "semi-sort cuckoo" best when FPP <= 3%	$1.44\ \log_2(1/FPP)$ best when FPP > 0.5%	$c\ [\ 1.44\ \log_2(1/FPP)\ ]$ where c is the number of bits per counter, e.g. 4
Availability	limited (as of early 2016) cpp java go	widely available	widely available

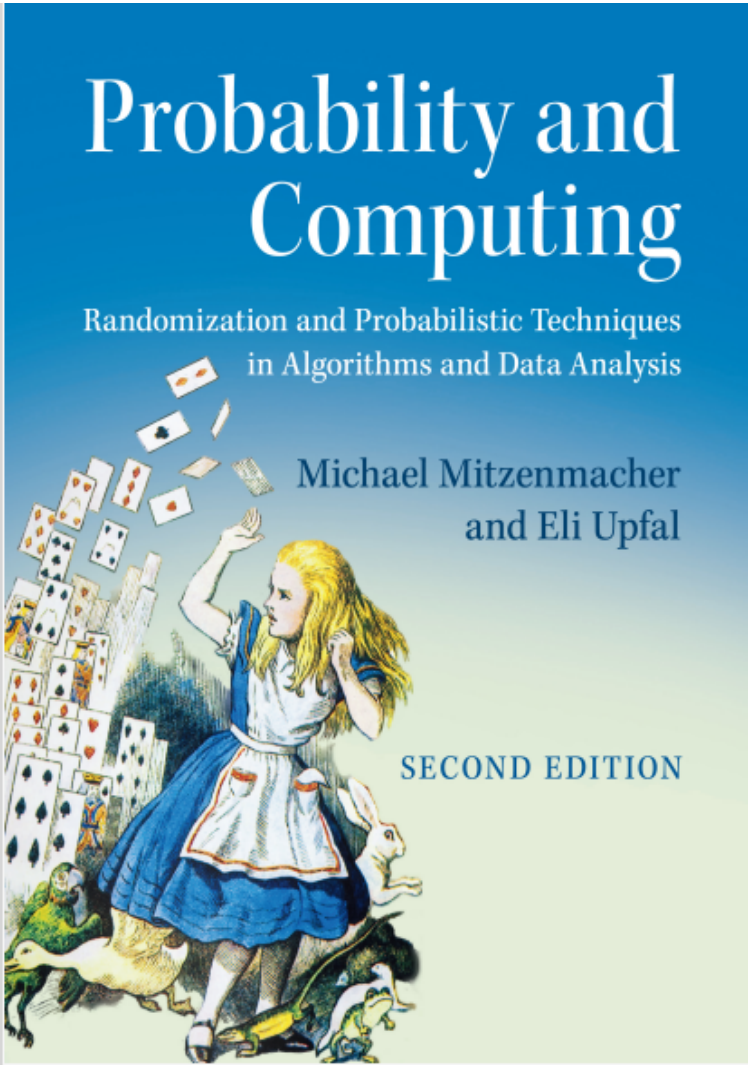
Interactive demo: Hash collision



Interactive demo: Design a bloom filter

```
Αρχείο  Επεξεργασία  Προβολή  Αναζήτηση  Τερματικό  Βοήθεια
$ !verbose on
Verbose mode: ON
$ !design 3000 0.001
   Bits   : 43133
   Memory : 42KB
   Hashes : 10
Num items : 0
 Fullness : 0%
False Pos ~ 0
$ !verbose off
Verbose mode: OFF
$ !import ../data/cia_words.txt
$ !check the hacker and the  hasker
Positives: 'hacker'
$ !stats
   Bits   : 43133
   Memory : 42KB
   Hashes : 10
Num items : 24
 Fullness : 0,278209%
False Pos ~ 2,76655e-23
$
```

Questions



One hash makes you member
And one hash makes you not

And the hash that std gives
you give false positives at all

If bits are low follow Alice