НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Data Structures and Organization

(p.6 – HashTables)

Yevhen Berkunskyi,

Computer Science dept., NUoS

eugeny.berkunsky@gmail.com

http://www.berkut.mk.ua

Hash Tables

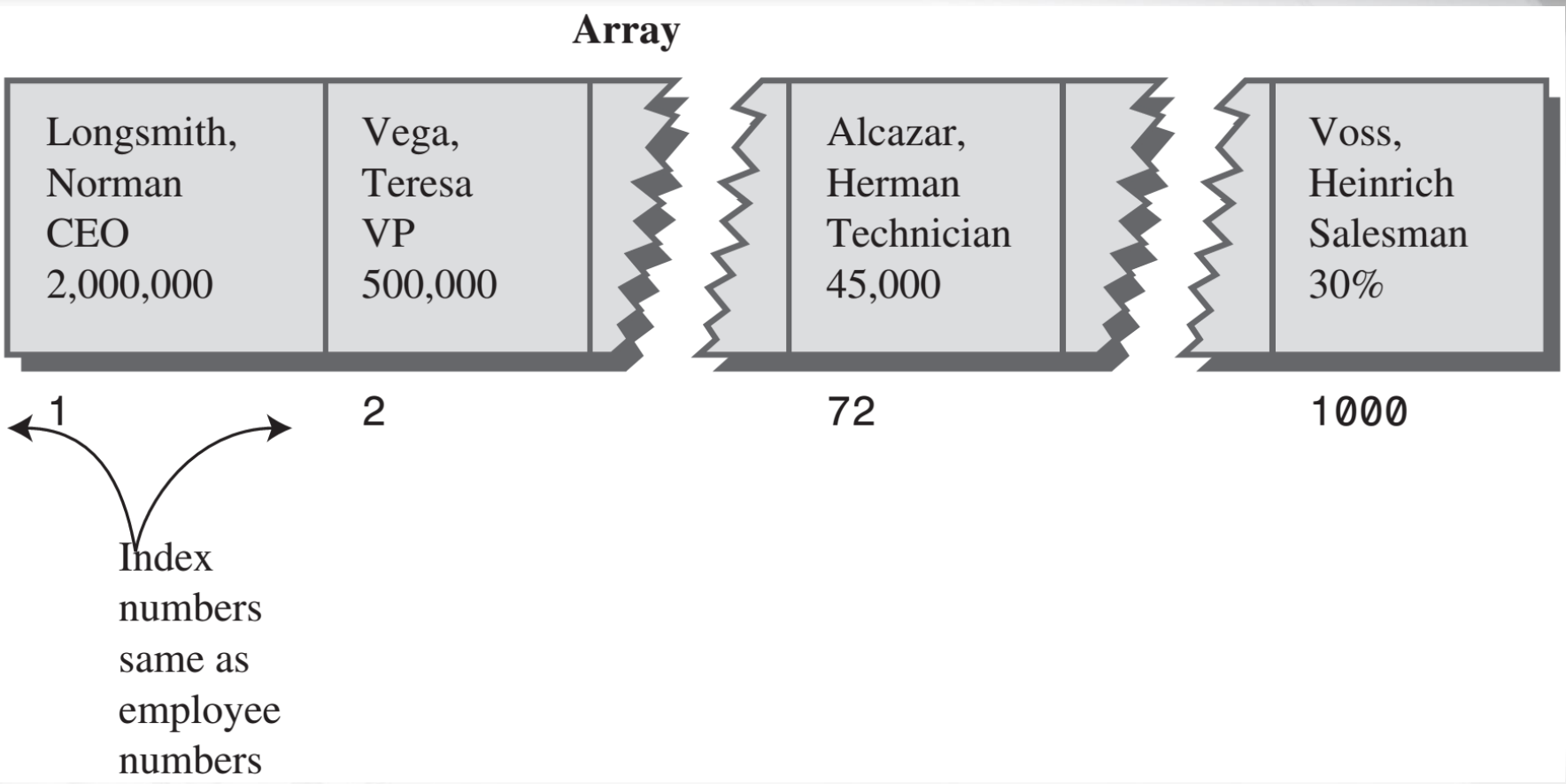
- A *hash table* is a data structure that offers very fast insertion and searching.
- When you first hear about them, hash tables sound almost too good to be true. No matter how many data items there are, insertion and searching (and sometimes deletion) can take close to constant time: O(1) in big O notation.
- In practice this is just a few machine instructions.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Introduction to Hashing

- Suppose you're writing a program to access employee records for a small company with, say, 1000 employees.
- Each employee record requires 1,000 bytes of storage.
- Thus, you can store the entire database in only 1 megabyte, which will easily fit in your computer's memory.
- The company's personnel director has specified that she wants the fastest possible access to any individual record. Also, every employee has been given a number from 1 (for the founder) to 1,000 (for the most recently hired worker).
- These employee numbers can be used as keys to access the records; in fact access by other keys is deemed unnecessary

Hashing


Array
Longsmith, Norman CEO 2,000,000
Vega, Teresa VP 500,000
Alcazar, Herman Technician 45,000
Voss, Heinrich Salesman 30%
1
2
72
1000
Index numbers same as employee numbers

A Dictionary

- Let’s say we want to store a 50,000-word English-language dictionary in main memory.
- You would like every word to occupy its own cell in a 50,000-cell array, so you can access the word using an index number. This will make access very fast.
- But what’s the relationship of these index numbers to the words? Given the word **morphosis**, for example, how do we find its index number?

Converting Words to Numbers

- ASCII code runs from 0 to 255, to accommodate letters, punctuation, and so on.
- There are really only 26 letters in English words, so let’s devise our own code, a simpler one that can potentially save memory space.
- Let’s say a is 1, b is 2, c is 3, and so on up to 26 for z.
- We’ll also say a blank is 0, so we have 27 characters.

Converting Words to Numbers

- **Adding the Digits**
  A simple approach to converting a word to a number might be to simply add the code numbers for each character. Say we want to convert the word *cats* to a number.
  First, we convert the characters to digits using our homemade code:
  c = 3
  a = 1
  t = 20
  s = 19
  Then we add them: 3 + 1 + 20 + 19 = 43

What is the problem?

Converting Words to Numbers

Multiplying by Powers

Say we want to convert the word cats to a number. We convert the digits to numbers as shown earlier. Then we multiply each number by the appropriate power of 27 and add the results:

3*273 + 1*272 + 20*271 + 19*270

Calculating the powers gives

3*19683 + 1*729 + 20*27 + 19*1

and multiplying the letter codes times the powers yields

59049 + 729 + 540 + 19

which sums to 60337

What is the problem now?

Hashing

What we need is a way to compress the huge range of numbers we obtain from the numbers-multiplied-by-powers system into a range that matches a reasonably sized array.

A similar expression can be used to compress the really huge numbers that uniquely represent every English word into index numbers that fit in our dictionary array:

arrayIndex = hugeNumber % arraySize;

This is an example of a *hash function*. It *hashes* (converts) a number in a large range into a number in a smaller range. This smaller range corresponds to the index numbers in an array. An array into which data is inserted using a hash function is called a *hash table*.

Collisions

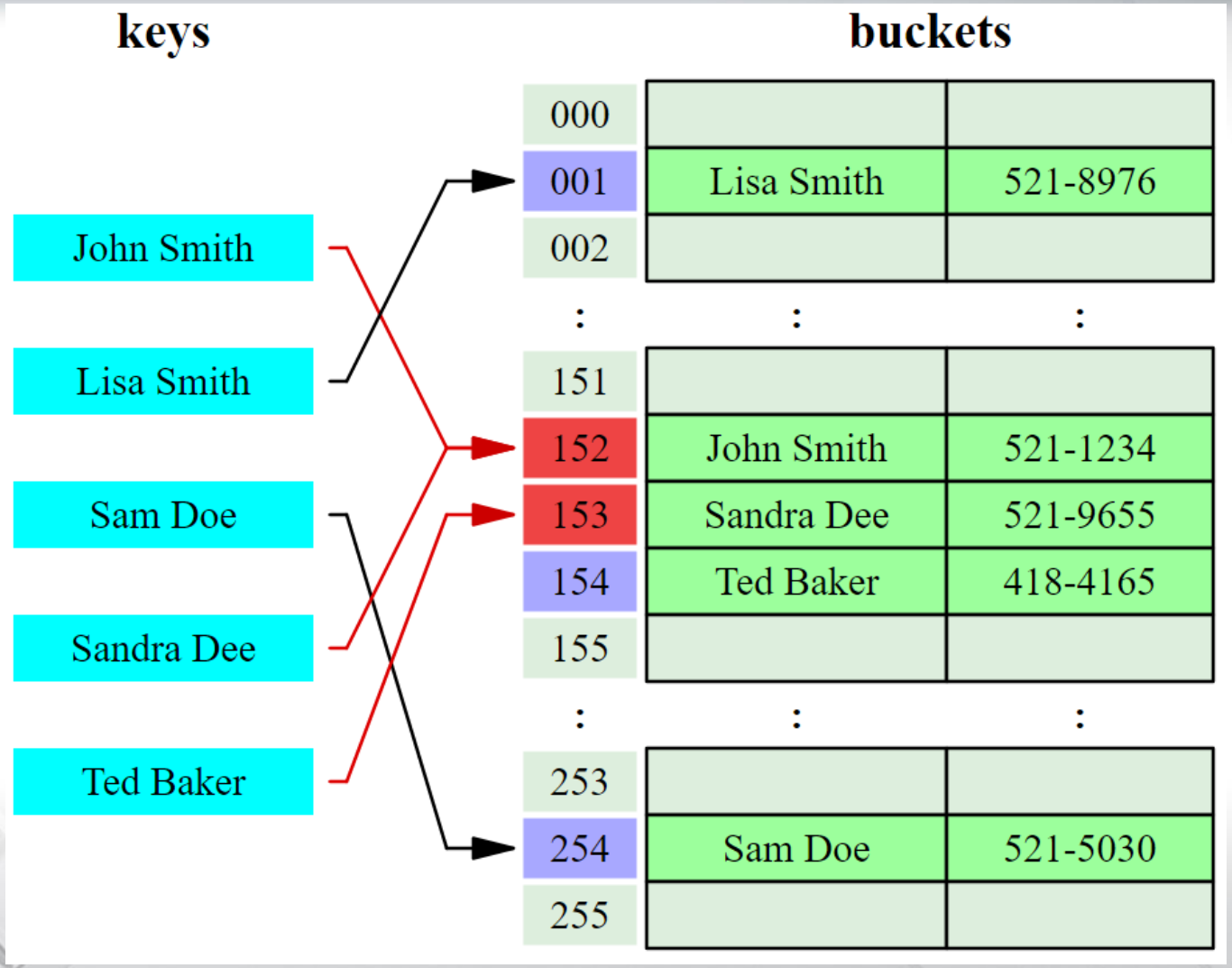
- We pay a price for squeezing a large range into a small one. There's no longer a guarantee that two words won't hash to the same array index.
- This is similar to what happened when we added the letter codes, but the situation is nowhere near as bad.
- When we added the letters, there were only 260 possible results (for words up to 10 letters).

Solutions

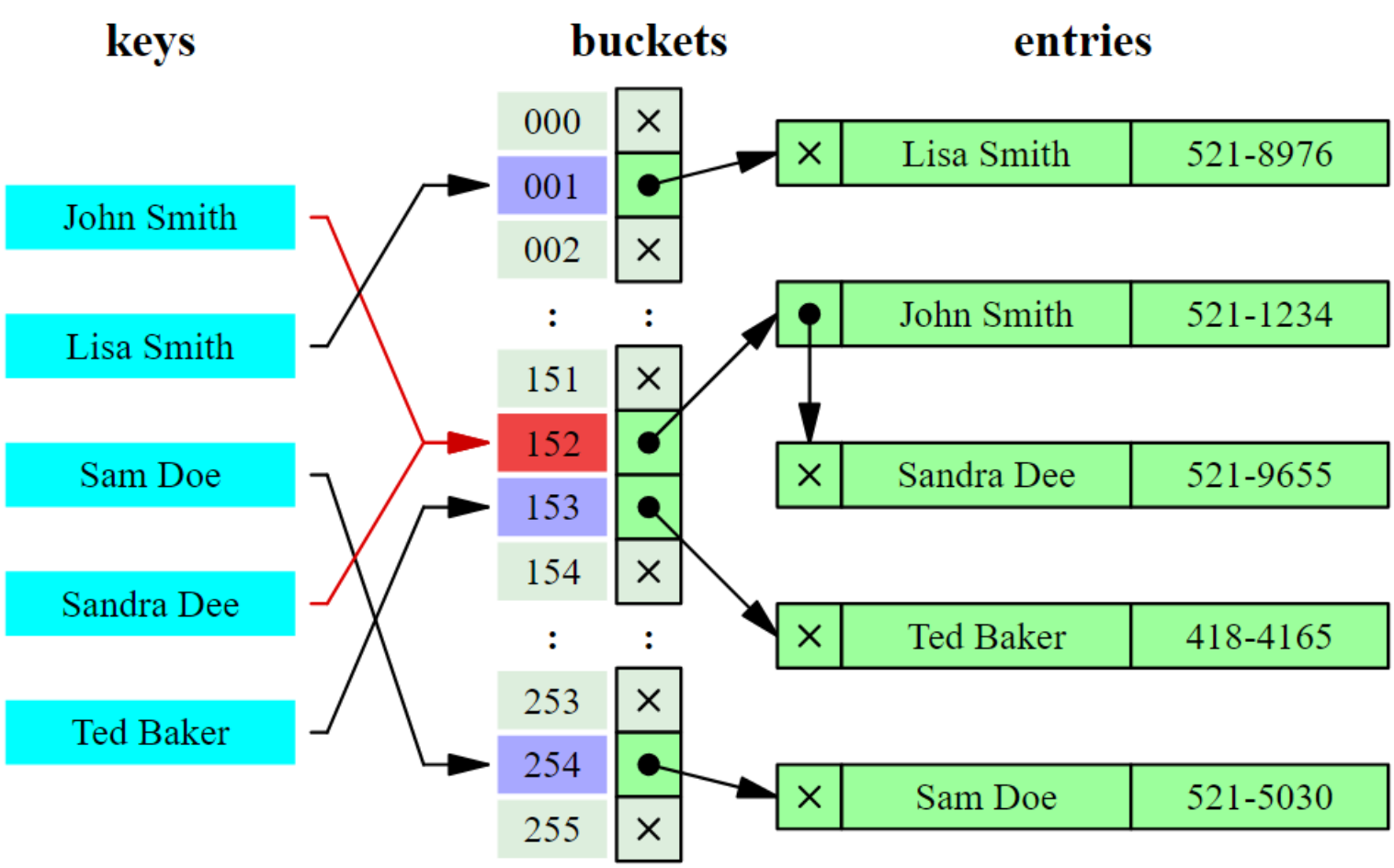
1) Open Addressing

2) Separate Chaining

Open Addressing



Separate Chaining



Load Factors

- The load factor (the ratio of the number of items in a hash table to its size) is typically different in separate chaining than in open addressing. In separate chaining it's normal to put N or more items into an N cell array; thus, the load factor can be 1 or greater.
- There's no problem with this; some locations will simply contain two or more items in their lists.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Example

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Questions?

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Data Structures and Organization

(p.6 – HashTables)

Yevhen Berkunskyi,

Computer Science dept., NUoS

eugeny.berkunsky@gmail.com

http://www.berkut.mk.ua