Indexing

ClassAlgorithmic Methods of Data MiningProgramM. Sc. Data ScienceUniversitySapienza University of RomeSemesterFall 2015LecturerCarlos Castillo http://chato.cl/

Sources:

- Slides by Silberschatz et al. 2006
- Fundamental data structures Wikibook

Index

INDEX		
INDEX		
Entries in heavy type refer to an	Anti-bacterial finish 64	
entire section on that subject;	Anti-static finish 64	sponge 492-3
entries in italics are sewing projects.	Appliqué	two-compartment 512-13
	lace trim 328	Rall stitch 26
	pockets 293	Ball-point needles
	renovating clothes 412	hand 17
A	Apron 504–5	machine 28
	Apron-length curtains 430	Band
Accordian alout 100	Armholes	mock turtleneck 224
Accordion pleats 182	altering fitting shell 101	turtleneck 224
Acedia fibre 57	taping 391	Bands
Adjustmente ese Alteratione	Arms	hems 317
Adjustments see Alterations	measuring children's length 400	mitring 322–3
hodice 409	Arrownead stitch	neckline
length 408	mand 147	limited-stretch strip 223
nettern (advanced) 97, 102	machine 156	neck and placket 222
armhole 101	Aud 12	shaped 221-2
back 100	AWITS	straight placket 222
hust 99		strin 221 223_6
neckline 98	-	zip insertion 225-6
shoulder 98, 100	R	shirred 265
side seam 102	D	Bartack
skirt back 103		hand 147
skirt front 102	Baby	machine 156
sleeves 101	patterns for 401	pocket reinforcement 294
trouser back 103	Back	Basket weave 59
trouser front 102	altering fitting shell 100	Basting see Tacking
trouser legs 103	Backstitch 136	Beach outfit 456–7
pattern (basic) 86-96	even 139	Beading needles 17
bust 91	half 140	Bedspreads 425–8
crotch seam 96	machine 151	estimating fabric requirements 425
hipline 94–95	pocket reinforcement 294	flounced
length 89–90, 190	Bag	description 425
waist 92–93	canvas hold-all 512-13	making 427
width 190-1	jewellery 491-3	aning 428
pleated garment 190-1	laundry 516	quined patchwork STO-TT
skirt 408	sewing 491, 493	description 425
trousers 409	shouldes FOG 7	making 428
waistline 408	shoulder 500-7	maxing 450

What is an index?

Why an index?











Sequential files for storing items (e.g. documents)

record-code (20 bytes, e.g. "REU20151014115203001") record-length (4 bytes) contents (*record-length* bytes))+

Problem: accessing a specific item requires to scan the entire file

Index

- Physical storage of data cannot be assumed to follow the order in which we want to access records
- Indices speed up data access
- Pairs of <search key, pointer>
- Ordered indexes (such as the index in a book)
 - Entries are sorted
- Hashed indexes
 - Entries are on a hash table

Example dense index

	Record code	Record number
	REU20151014115203001	1
	CNN20151014115491001	2
	CNN20151014115491002	3
	BBC20151015231704001	4
• •	•	

Record code	File number, Record number
REU20151014115203001	1, 1
CNN20151014115491001	1, 2
CNN20151014115491002	1, 3
BBC20151015231704001	2, 1

Is this index helpful? Yes, no, why?

Example sorted dense index

Record code	File number, Record number
BBC20151015231704001	2, 1
CNN20151014115491001	1, 2
CNN20151014115491002	1, 3
REU20151014115203001	1, 1

What can we do when the index becomes too big to fit in main memory?

Example sorted 2-level dense index

Record code	Index number
BBC20151015231704001	2
CNN20151014115491001	1
CNN20151014115491002	1
REU20151014115203001	1
Depart and (index 1)	Decord number

Record code (muex I)	Record number
CNN20151014115491001	2
CNN20151014115491002	3
REU20151014115203001	1

Record code (index 2)	Record number
BBC20151015231704001	1

. . .

What can we do when the first-level index becomes too big to fit in main memory?

. . .

Example sorted 2-level sparse index

Record code	Index number
BBC20151015231704001	2
CNN20151014115491001	1
REU20151014115203001	1
•••	
Record code (index 1)	Record number
Record code (index 1) CNN20151014115491001	Record number 2
Record code (index 1) CNN20151014115491001 CNN20151014115491002	Record number 2 3
Record code (index 1) CNN20151014115491001 CNN20151014115491002 REU20151014115203001	Record number 2 3 1

We know CNN20151014115491002 goes after CNN20151014115491001 and before REU20151014115203001, we don't include it in the first-level index if its index number is the same as before. Many other tricks are possible.

How many operations to find a record in a (first-level, dense) index?

food, 38 insect sting, 72 pollen, 72 skin, 70, 76 allergies and bronchial constriction, 72 Alligator juniper, 65, 116, 117 Aloe, 13, 15, 153, 172, 220, 222, 223, 224, 226, 227 Aloe vera, 13 A. barbadensis, 13 A. ferox, 13 A. perfoliata var. vera, 13 A. vulgaris, 13 Aloysia triphylla, 87 A. wrightii, 87 Altamisa, 87 alterative, 35, 80, 104, 213, 229 alveoli, 67, 101, 229, 232 Ambrosia ambrosioides, 39 A. artemisiifolia, 40 A. deltoidea, 40 A. trifida, 40 Ambrosia spp., 39 ameba, 66. See also Entamoeba histolytica ameba infection and Cypress, 66 amebiasis, 222, 229, 232, 235, 239. See also amebic infection; montezuma's revenge; traveler's diarrhea amebiasis and Desert barberry, 79 amebic infection and Sagebrush, 169 Tree of heaven, 189 American elders, 96 American Indian, 59, 90, 96, 106, 168

American licorice, 207 American pioneers, 169 ammonia and Yucca, 216 Amsinckia spp., 160 Anacardiaceae, 179 Anaphalis spp., 63 anaphrodisiac, 230 anaphrodisiac and Chaste tree, 45 Western black willow, 198 Anemone patens, 76 A. tuberosa, 75 angina pectoris, 230 angina pectoris and Puncturevine, 157, 222 anovulatory cycle, 230 anovulatory cycle and Chaste tree, 44, 227 Antelope horns, 16, 17, 18, 19, 85, 86, 123, 220, 222, 225, 226, 227 Antennaria spp., 63 anthraquinones and Aloe, 14, 15, 16 Copperleaf, 55 Senna, 172, 173 Syrian rue, 182 Yellowdock, 213, 214 antibacterial qualities of, Antelope horns, 18 Desert barberry, 79 Manzanita, 101, 127 Purple gromwell, 160 Tamarisk, 184 Western mugwort, 202 antibiotics and Candida overgrowth, 91, 192 Helicobacter pylori, 207 anticholinergic, 230 anticholinergic qualities of Baccharis, 21

Datura, 72

How do you find a word?

You go to the middle, look for the word there then go to the middle of the corresponding part ... until finding the word.

If there *n* words, you have multiplied *n* by $\frac{1}{2}$ until it became 1.

How many operations?

How many operations to find a record in a (first-level, dense) index?

food, 38 insect sting, 72 pollen, 72 skin, 70, 76 allergies and bronchial constriction, 72 Alligator juniper, 65, 116, 117 Aloe, 13, 15, 153, 172, 220, 222, 223, 224, 226, 227 Aloe vera, 13 A. barbadensis, 13 A. ferox, 13 A. perfoliata var. vera, 13 A. vulgaris, 13 Aloysia triphylla, 87 A. wrightii, 87 Altamisa, 87 alterative, 35, 80, 104, 213, 229 alveoli, 67, 101, 229, 232 Ambrosia ambrosioides, 39 A. artemisiifolia, 40 A. deltoidea, 40 A. trifida, 40 Ambrosia spp., 39 ameba, 66. See also Entamoeba histolytica ameba infection and Cypress, 66 amebiasis, 222, 229, 232, 235, 239. See also amebic infection; montezuma's revenge; traveler's diarrhea amebiasis and Desert barberry, 79 amebic infection and Sagebrush, 169 Tree of heaven, 189 American elders, 96 American Indian, 59, 90, 96, 106, 168

American licorice, 207 American pioneers, 169 ammonia and Yucca, 216 Amsinckia spp., 160 Anacardiaceae, 179 Anaphalis spp., 63 anaphrodisiac, 230 anaphrodisiac and Chaste tree, 45 Western black willow, 198 Anemone patens, 76 A. tuberosa, 75 angina pectoris, 230 angina pectoris and Puncturevine, 157, 222 anovulatory cycle, 230 anovulatory cycle and Chaste tree, 44, 227 Antelope horns, 16, 17, 18, 19, 85, 86, 123, 220, 222, 225, 226, 227 Antennaria spp., 63 anthraquinones and Aloe, 14, 15, 16 Copperleaf, 55 Senna, 172, 173 Syrian rue, 182 Yellowdock, 213, 214 antibacterial qualities of, Antelope horns, 18 Desert barberry, 79 Manzanita, 101, 127 Purple gromwell, 160 Tamarisk, 184 Western mugwort, 202 antibiotics and Candida overgrowth, 91, 192 Helicobacter pylori, 207 anticholinergic, 230 anticholinergic qualities of Baccharis, 21

Datura, 72

How do you find a word?

You go to the middle, look for the word there then go to the middle of the corresponding part ... until finding the word.

If there *n* words, you have multiplied *n* by $\frac{1}{2}$ until it became 1.

This is about *log*₂(*n*) operations.

Hashing

Reducing search time from $\log_2(n)$ to 1 operation

CNN20151014115491001	Record Code	RNumber
	BBC20151015231704001	4
	CNN20151014115491001	2
	CNN20151014115491002	3
	REU20151014115203001	1

k buckets



Example hash table for names



Hashing

- Invented in 1953
- Map a large space of keys to a much smaller space of buckets
- E.g. keys = all the possible names of people
- E.g. values = numbers from 1 to 1,000
- All buckets should be close to equiprobable



Example hash function



Problem: bucket space is too big

Example hash function (k=26⁶)



h("Maria Rossi") = "RSSMRA"

Problem: still too big

Typical trick: design a good hash coding, then apply modulo table size



Example hash function (k=1,000,000, to fit easily in main memory)



h("Maria Rossi") = numval("RSSMRA") mod 1000000

What is the problem with this hash function?

A simple hash function for strings

```
function hash(s, k) {
    int val = 0;
    for( int i=0; i<length(s); i++ ) {
        val = 13 * val + s[i];
    }
    return val % k;
}</pre>
```

Collisions

Keys (*n*) Buckets (*k*)



What shall we do about collisions?

If *n* < *k*, can collisions be completely avoided?

See https://en.wikipedia.org/wiki/Perfect_hash_function

What to do with collisions? Solution 1: overflow entries



By Jorge Stolfi (Own work) [CC BY-SA 3.0], via Wikimedia Commons

What to do with collisions? Solution 2: separate chaining



Back to our document system



Performance of hashing table

- Search
- Insert
- Delete

How many operations are required for any of these?

What factors make a hash table faster/slower?

Exercise

```
function hash(s, k) {
    int val = 0;
    for( int i=0; i<length(s); i++ ) {
        val = 13 * val + s[i];
    }
    return val % k;
}</pre>
```

Assume the value of s[i] = 1, 2, 3, 4 if s[i] is "a", "b", "c", "d"

Create a hash table of size 4 for elements "a", "ca", "ac", "ba"