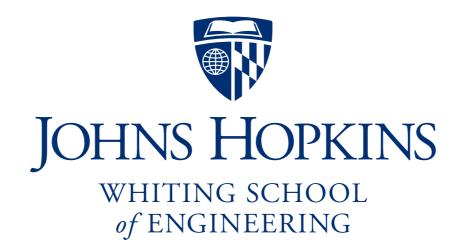
# Intro: Indexing

Ben Langmead



Department of Computer Science



Please sign guestbook (www.langmead-lab.org/teaching-materials) to tell me briefly how you are using the slides. For original Keynote files, email me (ben.langmead@gmail.com).

Imagine we have recorded the ages of many people; say, voters:

How many voters are aged 27? To find out, we have no choice but to scan n = 1M records

**Order** to the rescue

Index	Age
1	78
2	19
3	20
4	50
•	
500,000	54
500,001	50
500,002	19
500,003	77
•	
999,997	40
999,998	27
999,999	71
1,000,000	44

Example modeled on: Prezza, Nicola. Compressed Computation for Text Indexing. Diss. PhD thesis, University of Udine, 2016.

17	18
33	18
39	18
60	18

999,905	49
999,985	49
4	50
18	50

•

999,649	101
999,811	101
433,034	103
377,003	104

Suppose instead our list is ordered by age

How many voters are aged 27?

Binary search

More specifically?

2 searches, one for the first age-27 person, one for last

17	18
33	18
39	18
60	18

999,905	49
999,985	49
4	50
18	50

•
•
•

999,649	101
999,811	101
433,034	103
377,003	104

Simply *ordering* the data allows us to query it more efficiently

From n-item scan to two  $\log_2 n$  binary searches

Did it also improve our ability to *compress* the age data?

Yes; we now have "runs" of same value, monotonicity, etc

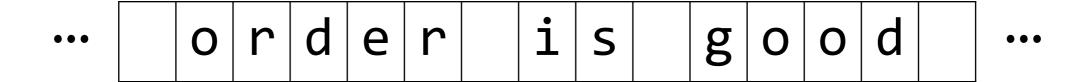


Grouping

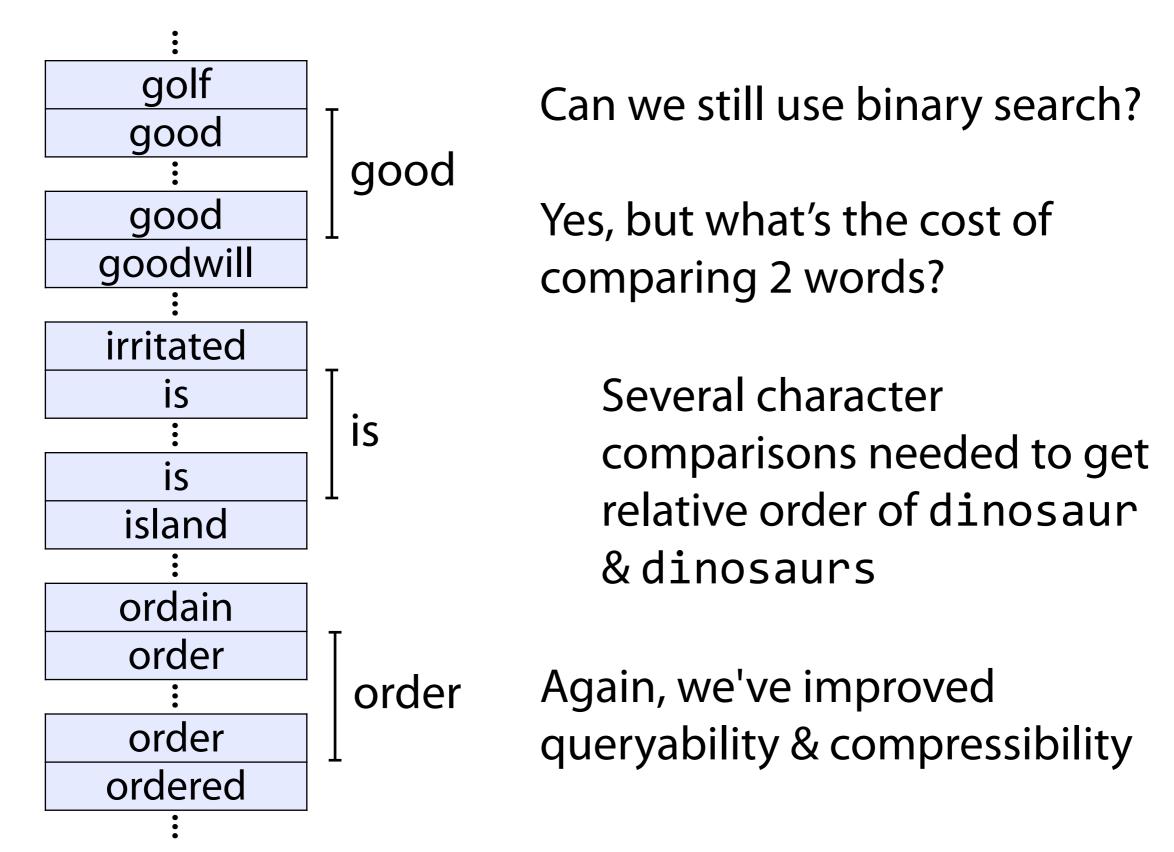
nest site hunting, 482-87 Macrotermes (termites), 59-60 honeypot ants, see Myrmecocystus male recognition, 298 hormones, 106-9 mass communication, 62-63, 214-18 see also exocrine glands mating, multiple, 155 house (nest site) hunting, 482-92 maze following, 119 Hymenoptera (general), xvi Megalomyrmex (ants), 457 haplodiploid sex determination, 20-22 Megaponera (ants), see Pachycondyla Hypoponera (ants), 194, 262, 324, 388 Melipona (stingless bees), 129 Melophorus (ants), repletes, 257 inclusive fitness, 20-23, 29-42 memory, 117-19, 213 information measurement, 251-52 Messor (harvester ants), 212, 232 intercastes, 388-89 mind, 117-19 see also ergatogynes; ergatoid queens; gamergates Monomorium, 127, 212, 214, 216-17, Iridomyrmex (ants), 266, 280, 288, 321 292 Isoptera, see termites motor displays, 235-47 mound-building ants, 2 juvenile hormone, caste, 106-9, 372 multilevel selection, 7, 7-13, 24-29 mutilation, ritual, 366-73 kin recognition, 293-98 mutualism, see symbioses, ants kin selection, 18-19, 23-24, 28-42, 299, Myanmyrma (fossil ants), 318 386 Myopias (ants), 326

Ordering

We are working with a text. We want to know if some word occurs. The text is big but an excerpt is:



Ordering words alphabetically: good < is < order



Queries only on words is limiting

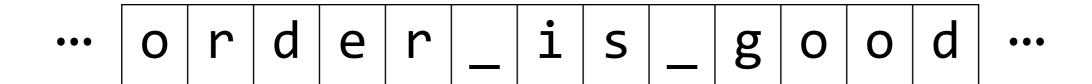
Texts might not consist of words e.g. DNA

Word matches might not be the right query

e.g. autocomplete

e.g. inexact matching

What if we'd like to be able to query any substring?



Use underscore (\_) for space, assume it comes first alphabetically

Put all suffixes in order...

```
good...
 is_good...
d...
der_is_good...
er_is_good...
good...
is_good...
od...
ood...
order is good...
r_is_good...
rder is good...
s_good...
```

(This is just the relative order of the order\_is\_good suffixes)

Can we use binary search?

Yes; still might need several character comparisons to get relative order of suffixes

## Motivating questions

How do we measure the amount of redundant *information* in a string?

How do we represent strings so that redundant information takes *minimal space*?

How can orderings "*reveal*" structure and make strings compressible?

How can ordering make strings fast to **search**, faster than binary search?