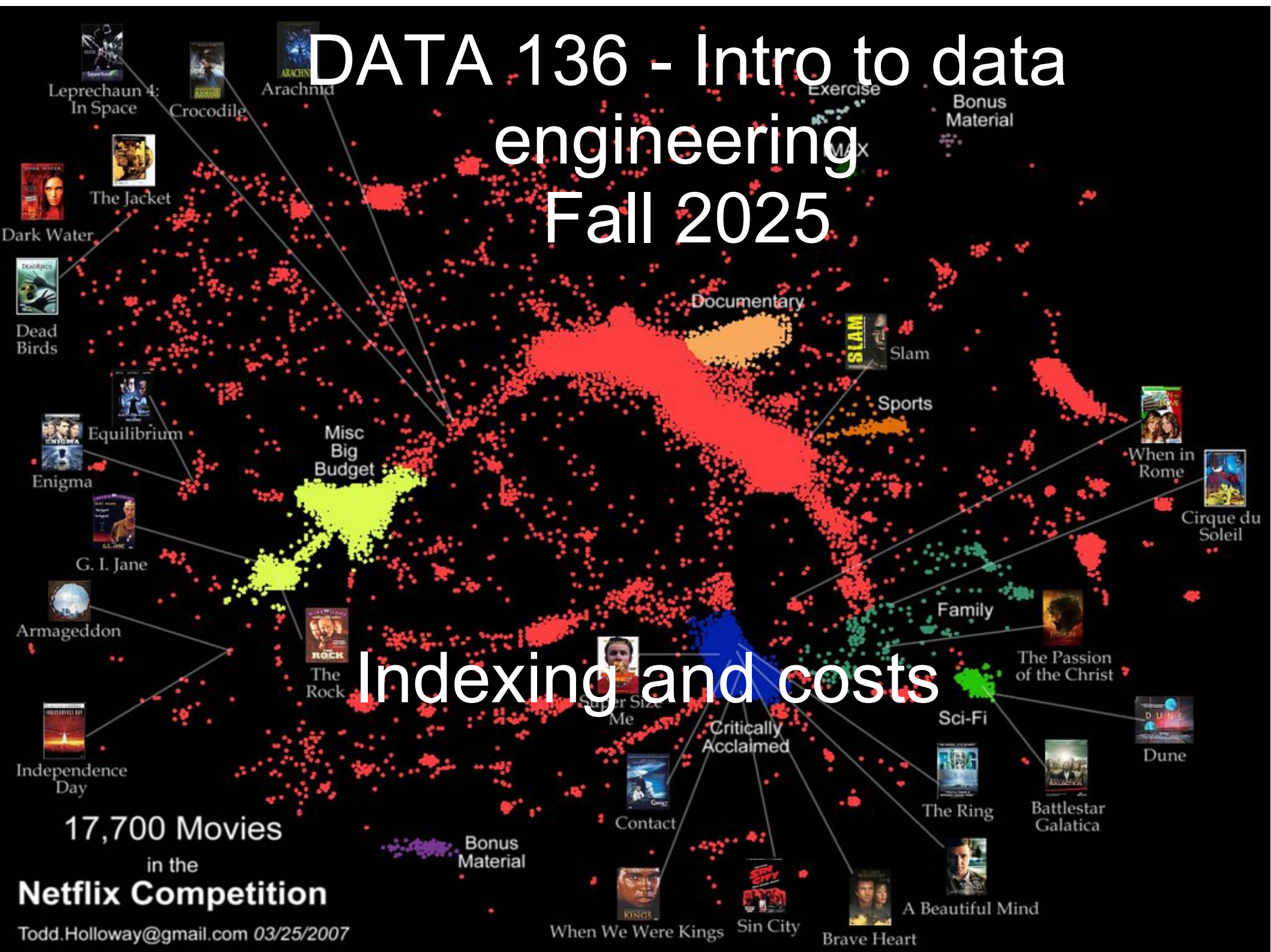
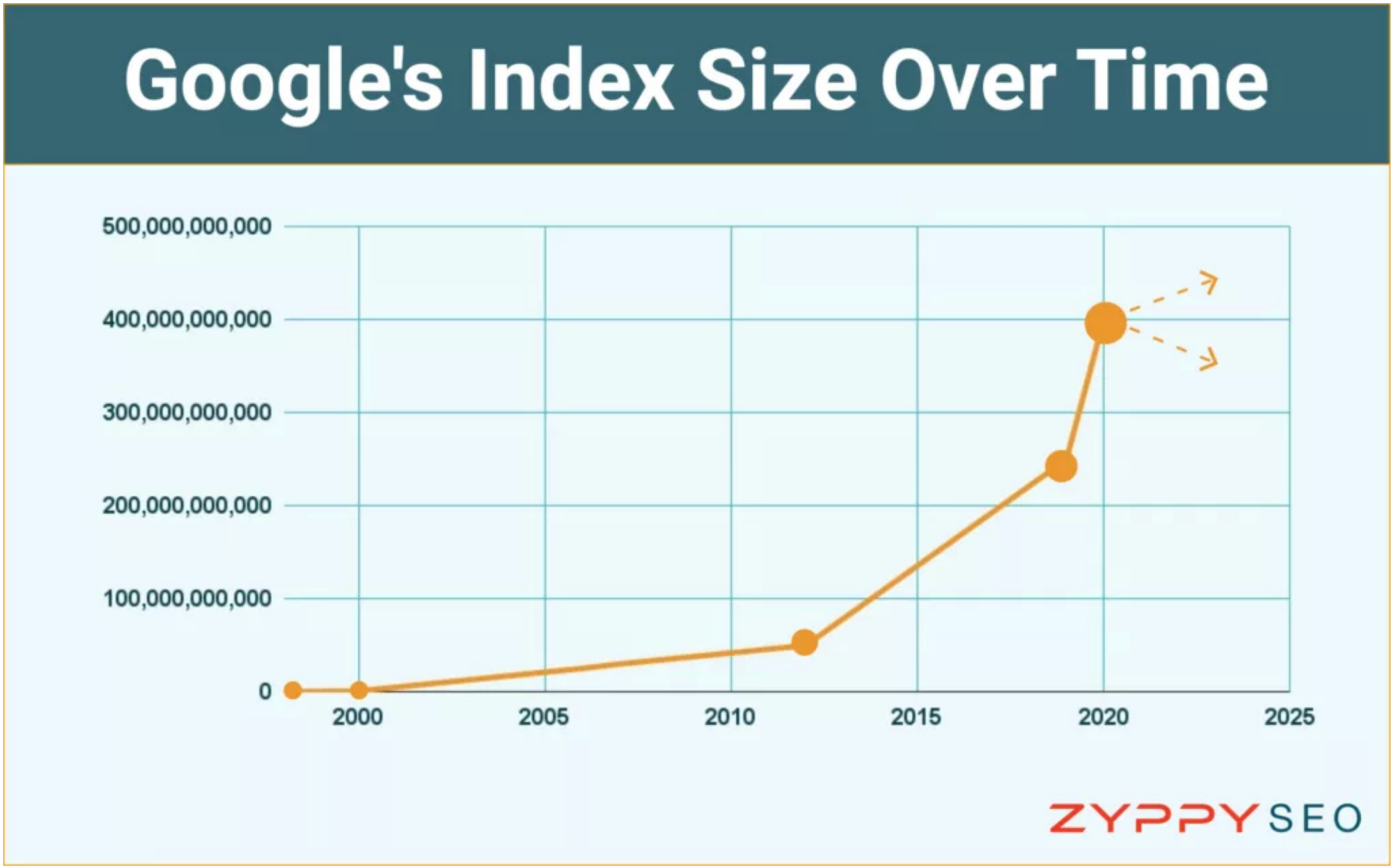
DATA 136 - Intro to data engineering Fall 2025

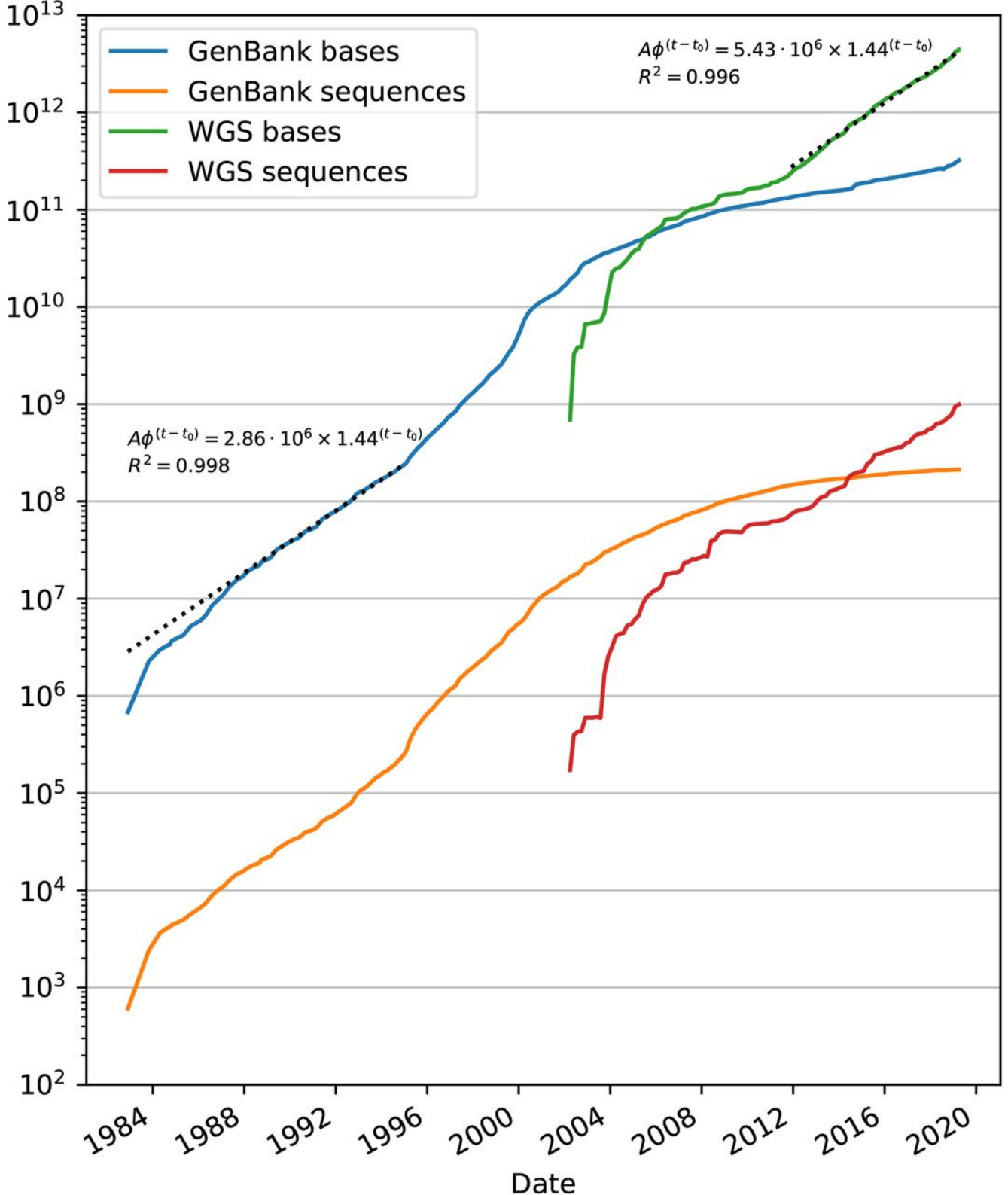


Indexing and costs

Motivation: search is getting harder



https://zyppy.com/seo/google-index-size/



Genome database searching is getting harder, too.

Marti (2019)

https://doi.org/10.1101/653592

How many steps to look something up?

How many steps to look something up?

Suppose I have a pile of books, in random order, and where my only access is by choosing a book (randomly).

Contrived? Yes...

How many books do I need to look at on average in a pile of size N to find a specific book (the one with the dedication from my high school biology teacher) ?

pseudocode for random search (an inefffective search strategy)

```
while sample_book != target_book:
        sample_book = get_new_book()
```

What happens if it can't find my book?

pseudocode for random search (with MAXTRIES)

```
sample_book = get_new_book()
n = 0
while sample_book != target_book and n<MAXTRIES:
        n +=1
        sample_book = get_new_book()
```

How many steps to look something up?

Suppose I have a pile of books, in random order, but I can make sure I never check the same book twice

How many steps to look something up?

Suppose I have a pile of books, in random order, but I can make sure I never check the same book twice

My number of books to check is Uniform (1/N) which has a mean of N/2

```
while sample_book != target_book:
        sample_book = get_new_book()
```

How many steps to look something up?

What if the books are sorted?

How many steps to look something up?

What if the books are sorted?

Binary search:

1. Look in the middle of the bookstack
2. compare the call number to target
3. If this is your book, terminate
4. else look in the upper (lower) half of the bookstack and repeat step 1.

How many steps?

How many steps to look something up?

What if the books are sorted?

Binary search:
1. Look in the middle of the bookstack
2. compare the call number to target
3. If this is your book, terminate
4. else look in the upper (lower) half of the bookstack and repeat step 1.

How many steps?

```
def bisect(MIN, MAX, target):
     n_book = get_nearest_book( (MAX + MIN) / 2)
     if n_book.index < target_book.index:
          bisect (MIN, (MAX+MIN)/2, target)
     if n_book.index > target_book.index:
          bisect ((MAX+MIN)/2, MAX, target)
```

How many steps to look something up?

What if the books are sorted?

Binary search:

1. Look in the middle of the bookstack
2. compare the call number to target
3. If this is your book, terminate
4. else look in the upper (lower) half of the bookstack and repeat step 1.

How many steps?

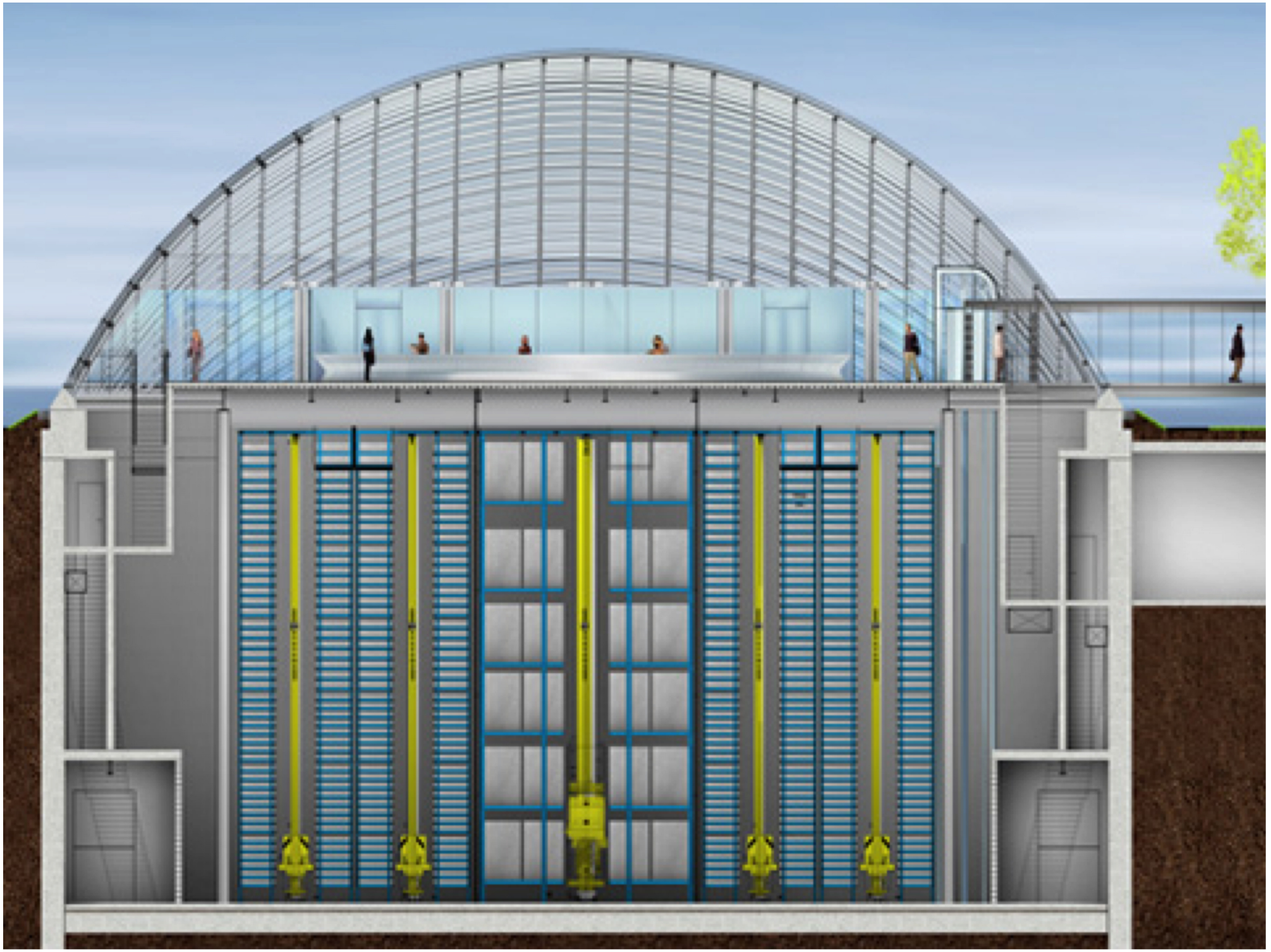
cieling($\log_2$ (N))

```
def bisect(MIN, MAX, target):
     n_book = get_nearest_book( (MAX + MIN) / 2)
     if n_book.index < target_book.index:
          bisect (MIN, (MAX+MIN)/2, target)
     if n_book.index > target_book.index:
          bisect ((MAX+MIN)/2, MAX, target)
```

How does Mansueto retrieve books?

Imagine that the location of every book is known. Retrieving a book has just two steps. Find out the location, and go to the location and get the book.

How does the cost of this procedure depend on the number of books in the library?



Big-O notation

- Way of expressing the functional dependence, in the limit of very large N, of **the number of computing operations** to achieve an analytical result on the sizes of the query (k) and of the database (n)
- Some analytical operations may become easier with larger query sets, like birthday problem or "crack the first password"
- Describes the behavior of the performance function in the limit of large N and large k.

Amortizing indexing costs

cat database.csv | grep -i "Harry Potter and the Plot Hole"

We can solve this (responding to a search query) faster by an initial investment in an index.

Lookup cost = cost of indexing + k cost per query

`Wk4-indexing.py`

What tricks do we have?

Type	setup	query time
Unordered list (naive search)	None	k O(N)
Ordered list	k O(N log(N))	k O(log(N))
Data at known address	O(N)	O(1)