$\log_k(N) = \frac{\log_2(N)}{\log_2(k)}$}

Index improves by constant multiple
$N$ records per page

Naive Binary Search $O(\log_2 N)$

With Index Pages $O(\log_{k} N)$
Idea 1: 2 (or more) separate copies of the index
pro: Easy
con: 2 copies of all data

Idea 2: Hierarchical sort
pro: Saves space
con: Limited query support
Idea: 2nd & subsequent indexes
Store only unique Id of records

Pro: Less Space (so maybe faster)

Con:
One Index for each attr

Easy

Space hog

Hierarchical Indexes

Saves space
Still Easy

Secondary Indexes

Saves space
Can be faster

Limited Query Support

Often need to load more pages
Given a query for rank=$\land$ age=$\land$

**Pro**

**Con**

Option 1: Pick an index (either rank or age)

Option 2: Scan the full data

Option 3: Find all keys for rank, for age
+ compute intersection

Option 4: Hierarchical Index

Question for Piazza
A

B

new records

D

E = B + D

F

G

H

I = D + H

J = A + I

LSM-Tree