Opsamling på HashMap

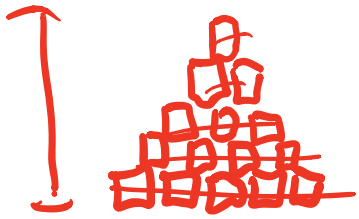
Rekursion ~ Codingbat ~ triangle

→ Parvis

→ løsninfra stud

→ Pyton tutor

→ debugger

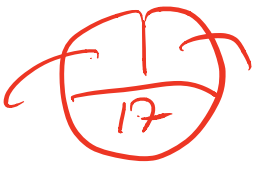


Søgetræ

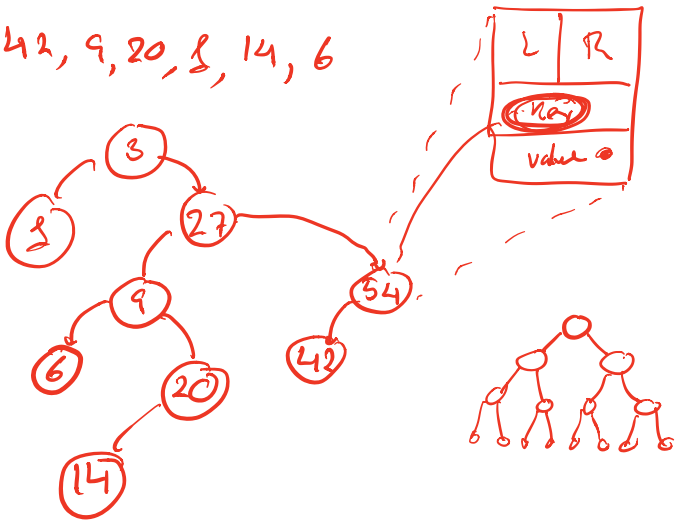
- operationer som Hashmap
- gennemgå (i tegninger)
  - søg
  - indsæt.

(• se på en animation
  • prøv (parvis) at gennemføre sletning)

~ ◦ —



3, 27, 54, 42, 9, 20, 6, 14, 6



Sortering e "store O"

1). Parvis: prøv at skitse hvordan man kan sortere et array af int

2). prøv at tælle hvor mange gange i sammenligner 2 tal (undo 8)
→ lav tabel

n	samlet
2	1
5	20 ,25
7	42
n	n·(n-1)

n	n·(n-1)
10	90
100	9900
1000	999000
10000	99990000
1000.000	999999000000
	32.000.000

Quicksort

1) Vælg et tal p i array A

2) Løb A igennem så den er

3) Gentag for hver af de to "halv dele"

store O af en algoritme

en måde at sige noget om hvor hurtig en algoritme er, afhængigt af problemets størrelse

→ Problem størrelse
- længde af array
- antal knuder i træ
- en parameter til en metode

→ Det vi "tæller"
- antal assignments
- antal array opslag
- antal knuder vi kikker på

Øvelser i grp

→ Hvor mange array opslag skal der laves før at finde det mindste tal i array?

→ Hvor mange opslag skal der laves for at se om 217 findes i et array?

→ Hvor mange gange kan du

- Dividere 41274287 med 10 for du kun har 1 tilbage?
- Dividere 32 med 2 for du kun har 1 tilbage
- Dividere N med 10
- Dividere N med 2

→ Hvormange gange skal du fordoble 2 for at res bliver

- over 1000?
- over 1000.000?
- over 1000.000.000?

Hvor god er quicksort sml. med bubble sort

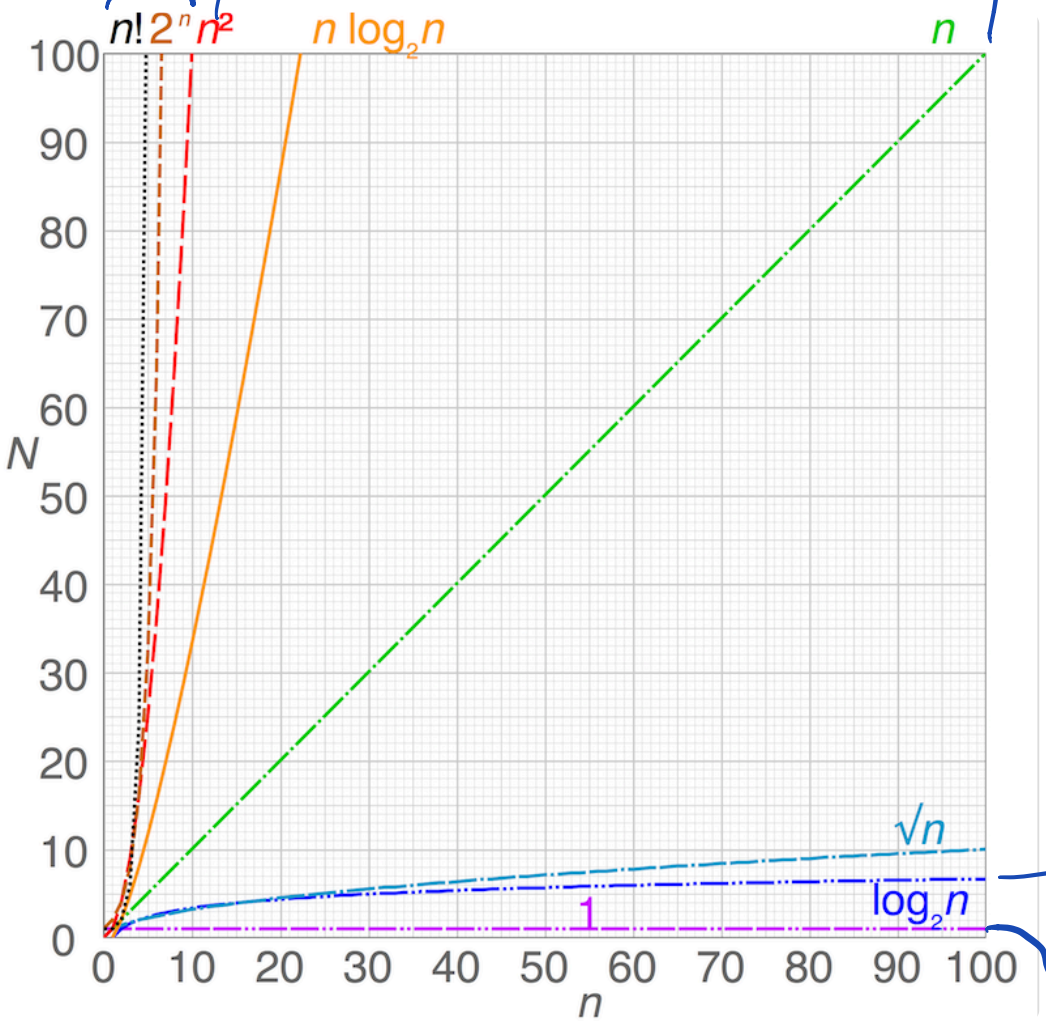
n	Bubble	Quicksort
10		
100		
1000		
1000.000		

basis for kryptering

bubblesort

quicksort

lede et array igennem



søge i tree

HashMap

String , Object

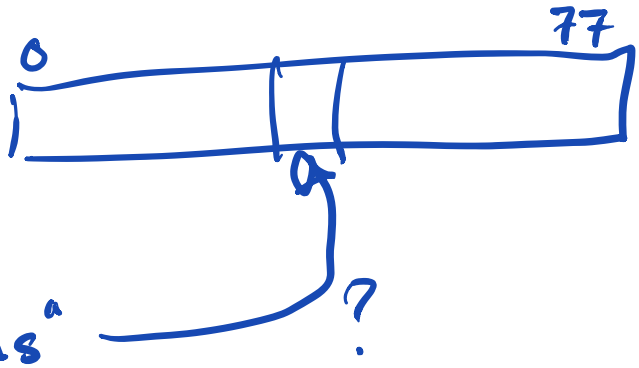
↓ Key ↓ Value

Hashmap

.put (key , value)

.get (key) → value

.includsKey(key) → boolean

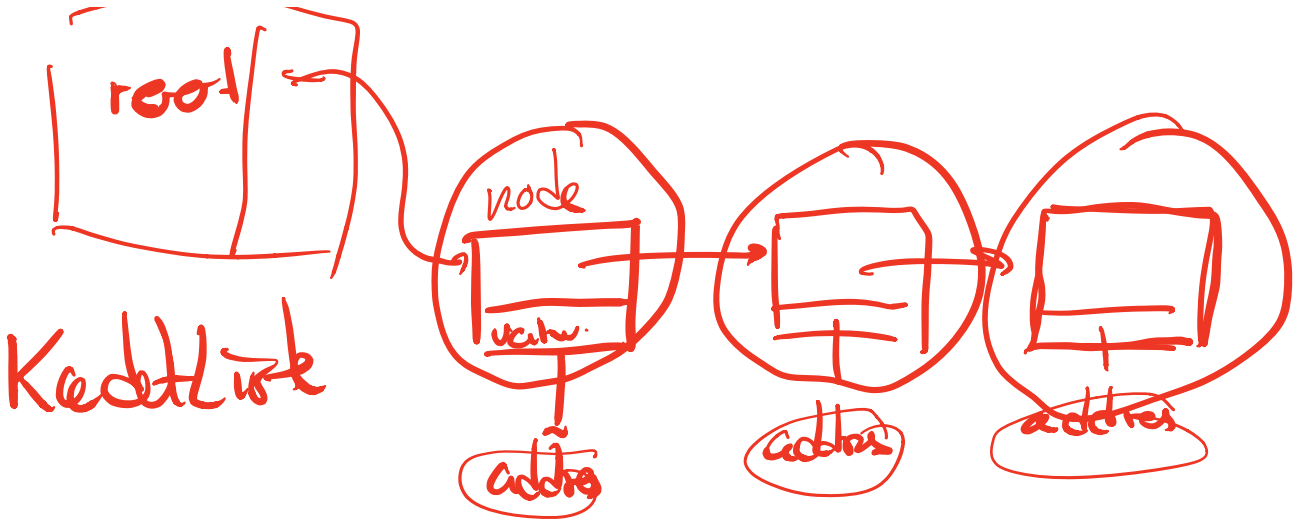


"Jens".hashCode() → 2 , 0 , -12

437219752

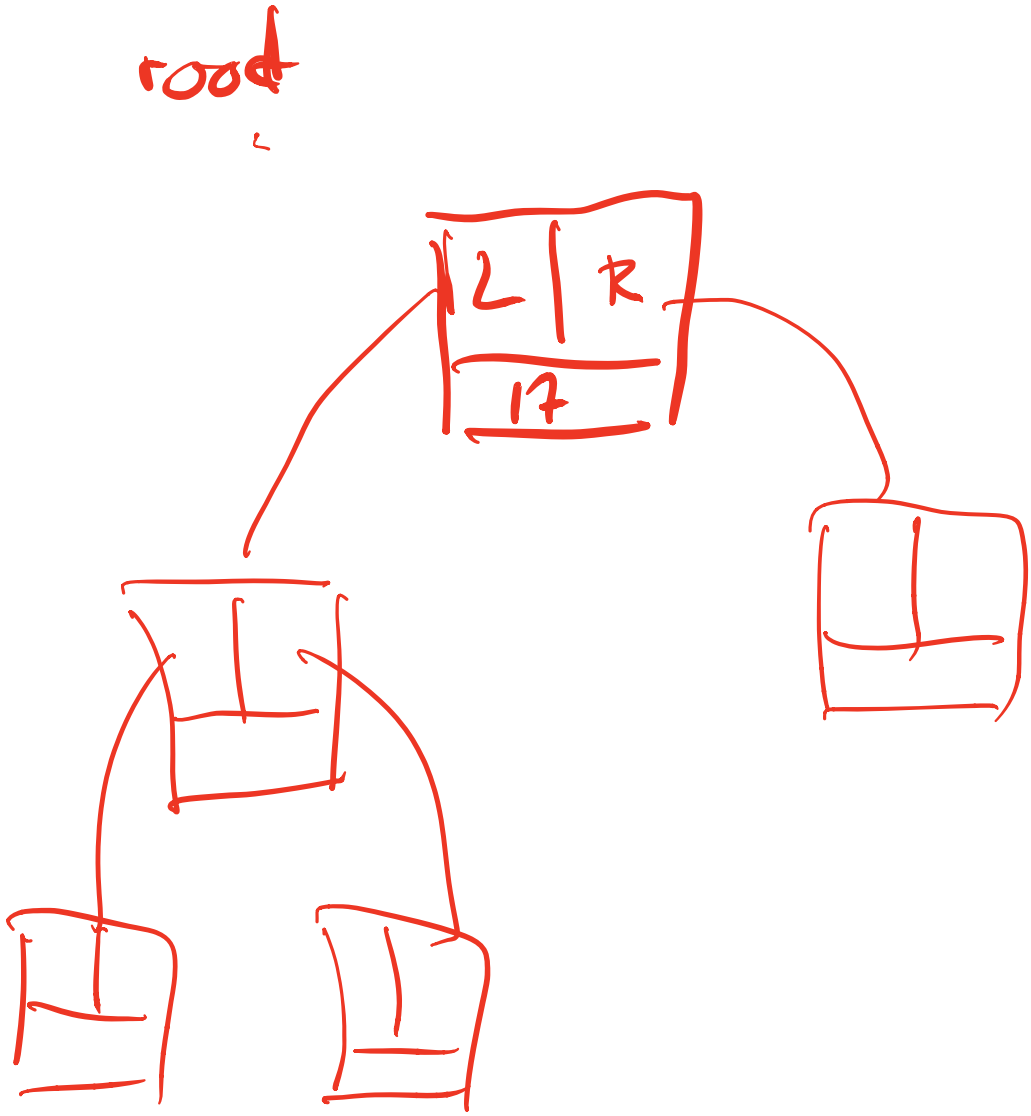
→ Math.abs( )

→ %.78




```
contains(Obj elem, Node head){
    if ( node == null ) return false;

    if (node.value == elem)
        return true;
    return contains(elem, head.next)
```


root
L
R
17