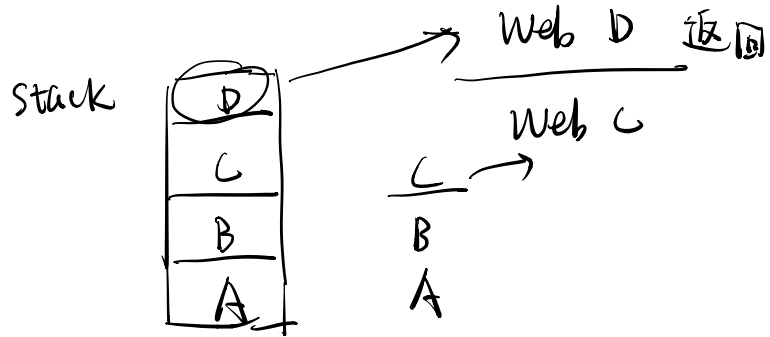


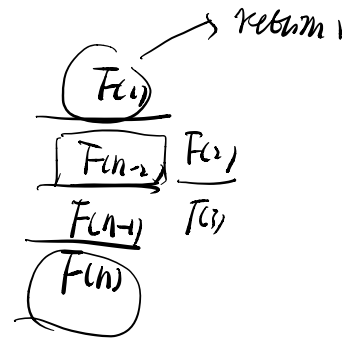
A → B → C → D



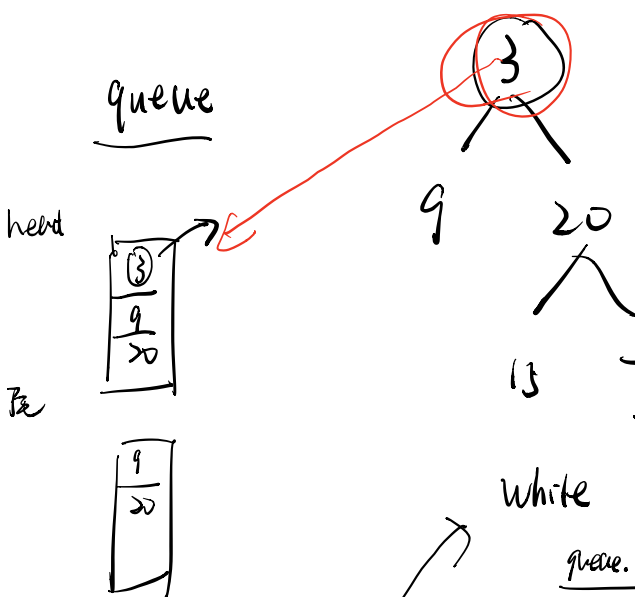
$$F(n) = F(n-1) + F(n-2)$$

1 1 2 3 5

系统栈



$$F(n) = F(n-1) + F(n-2)$$



BFS

① 分层次记录

遍历顺序

[3]
[9, 20]
[13, 7]

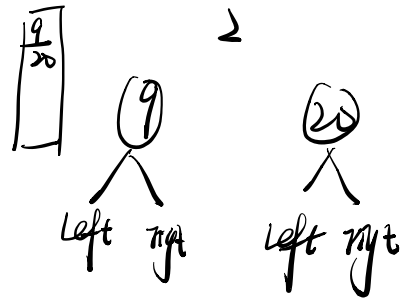
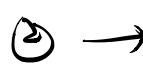
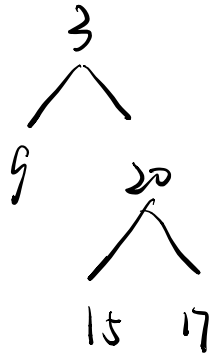
while !empty:

queue.size = node (2) size

for i in range(size)

pop() = node
tempList.append

if node.left
 push
 if node.right
 push
 template



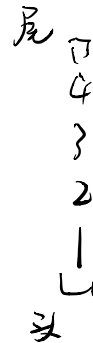
[3 , 9 , 20 , 15 , 17]

Stack

[1 2 3 4]

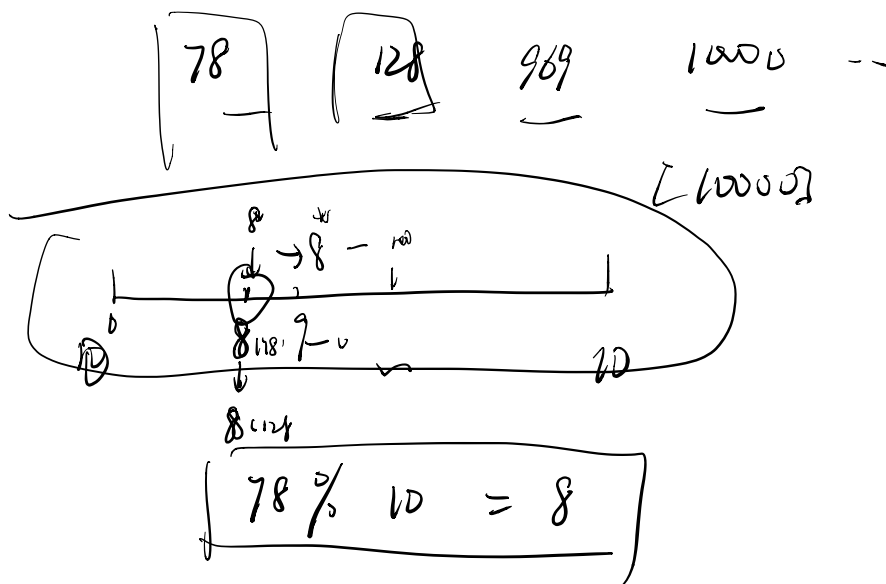
list.append (val)

list.pop()



↓
Hash
↓

散列值


$$\{K:V, K:V, K:V, \dots\}$$

$index_1$ $index_2$

{ num : \rightarrow index } $[index_1, index_2]$
 target - (num) \rightarrow in dict : index

2 7 11 15 ↗
 4

[2 11 15 7]
 =

data[temp] = 0

dict = { 2:0, 11:1, 15:2, 7:3 } i=3

temp = target - 2 = 7
 9

[0, 3]

[0, 0]

-2

-6

9-7 = 2

key → [.....] key

那么

eat

ten

vi

every words

ten

def

eat

ten

are

eat \rightarrow ['e', 'a', 't'] $\xrightarrow{\text{sort}}$ ['a', 'e', 't']
 \downarrow
 str \Rightarrow "aet"

eat \Rightarrow aet

tea \Rightarrow aet

aet

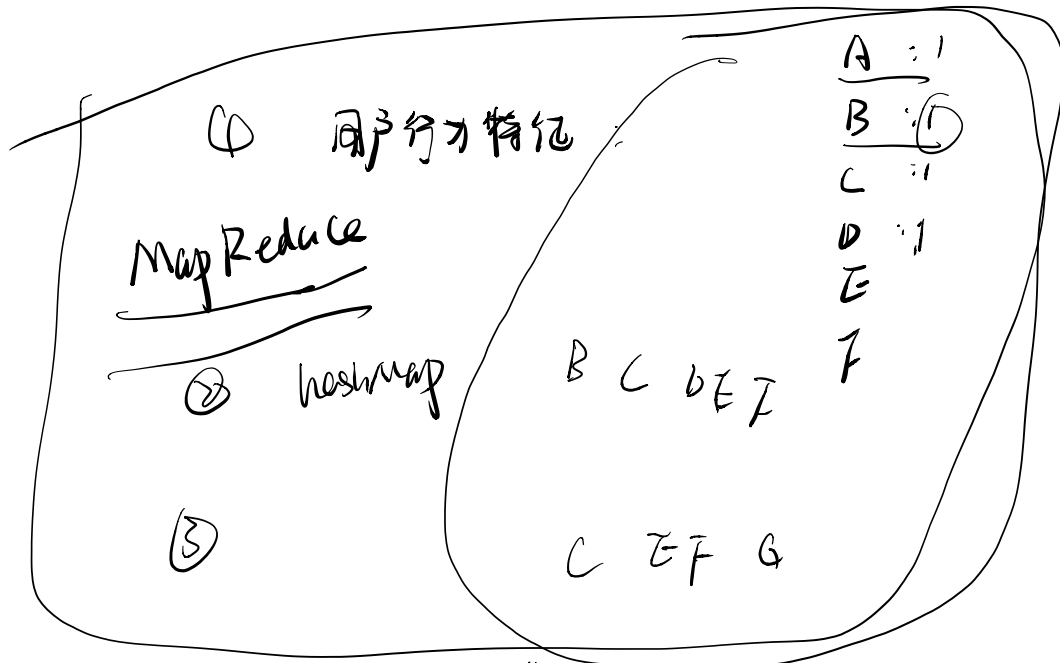
key \Rightarrow [eat, aet]

dict =

{
 key : str
 value : []
 }

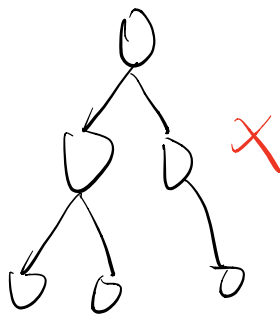
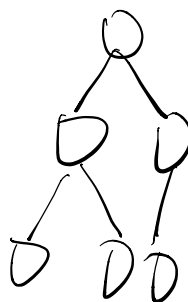
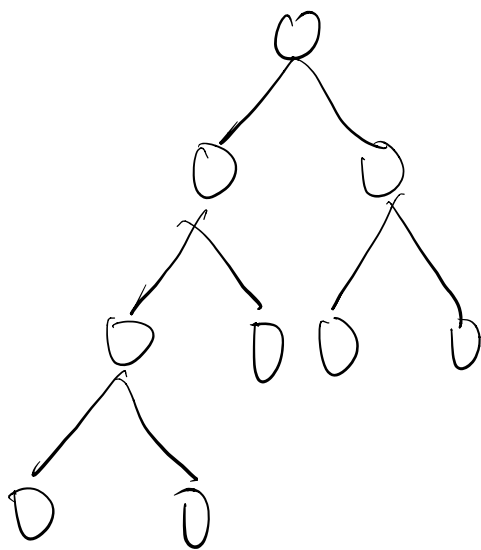
sde < 数据处理

FB

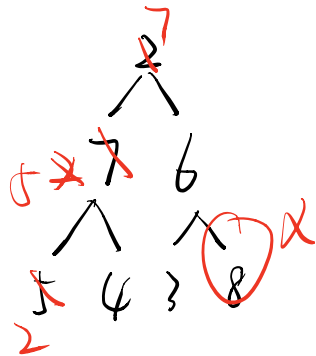
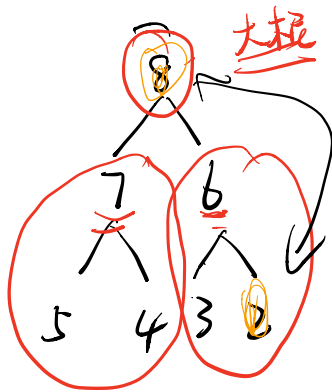




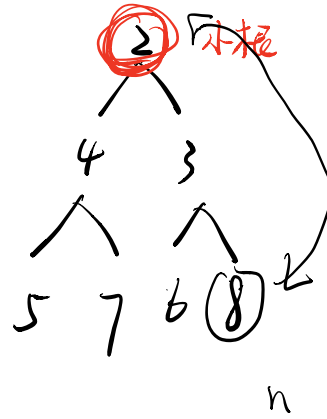
哪一篇整理最高



大根 Root

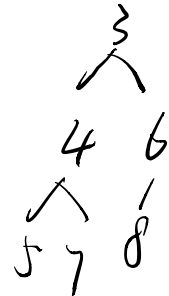
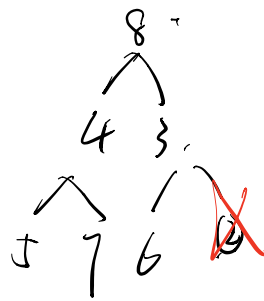


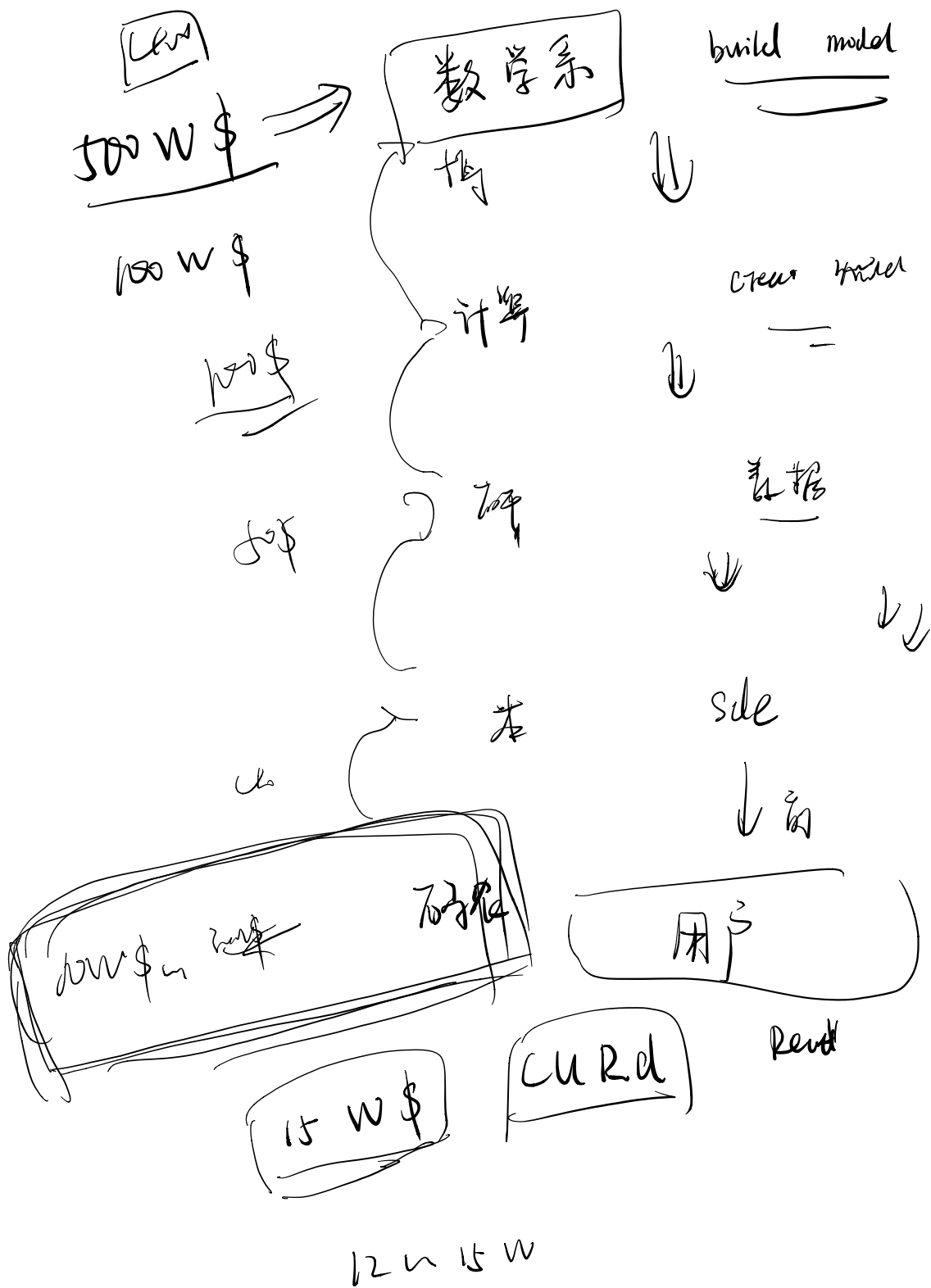
小根



pop() \Rightarrow 2

$\log(n)$

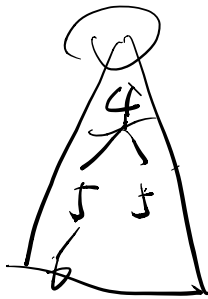




Heap

Top K

堆 4



heap top



在线算法:

1 2 3 4 5 6 7 8 10 12 15 6 27
Top 2
35 56 67

离线:

数据一旦给出不变

IDW 个数

Top 100

queue / stack 概念
hash map 多 辅助

Heap Topk 离/在