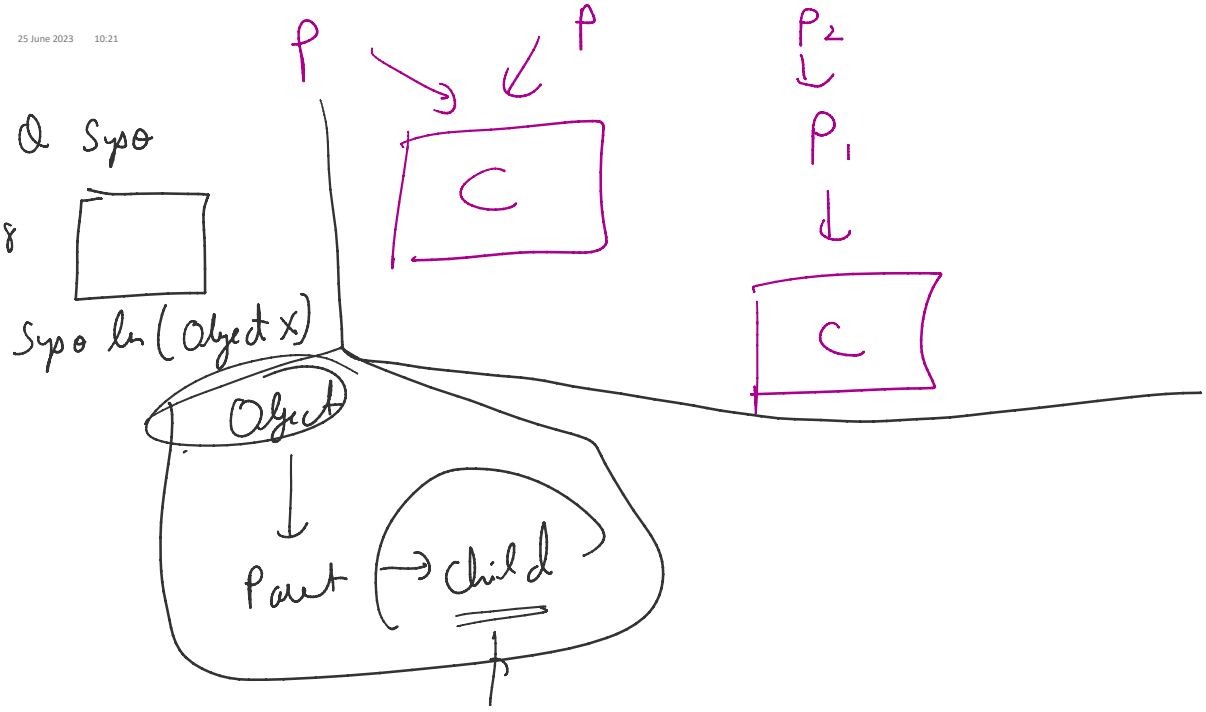
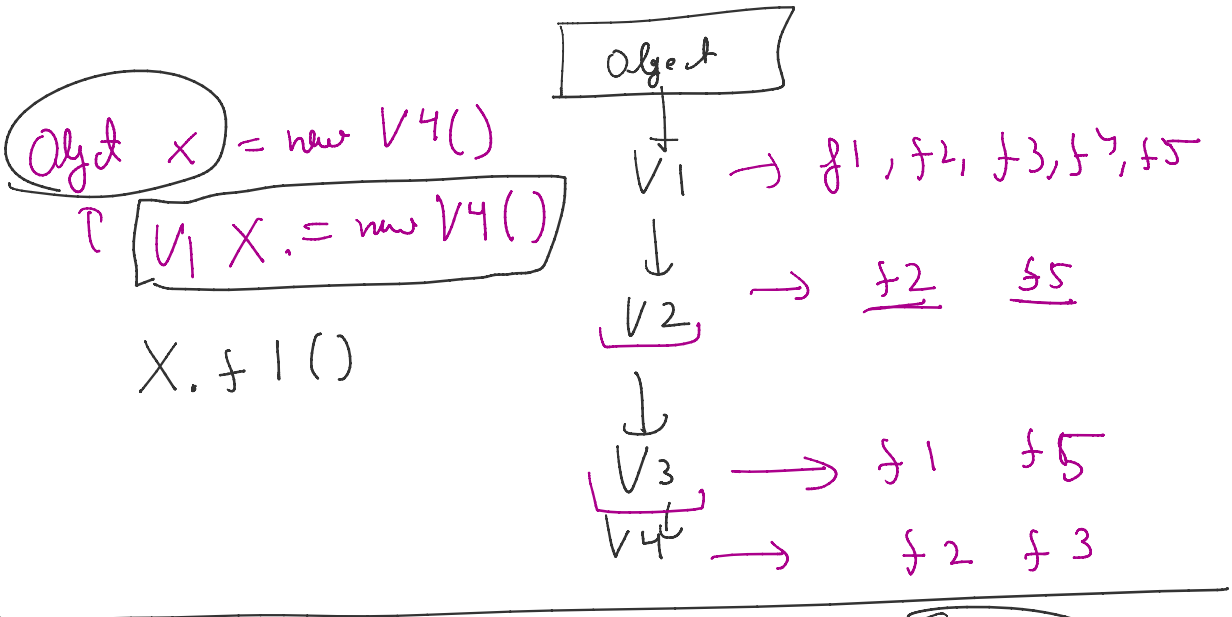
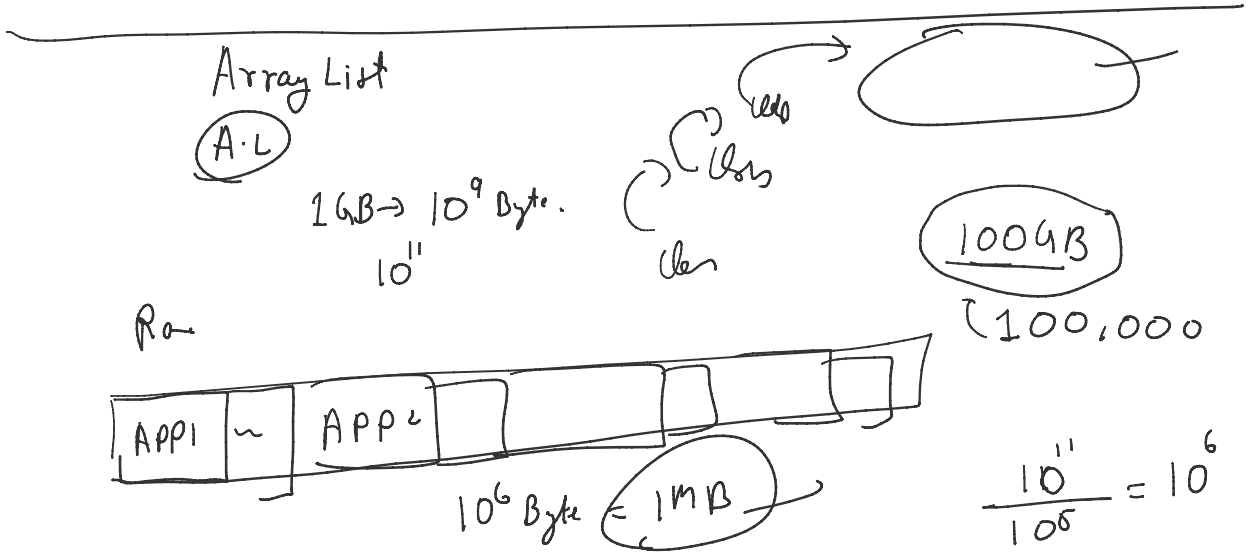
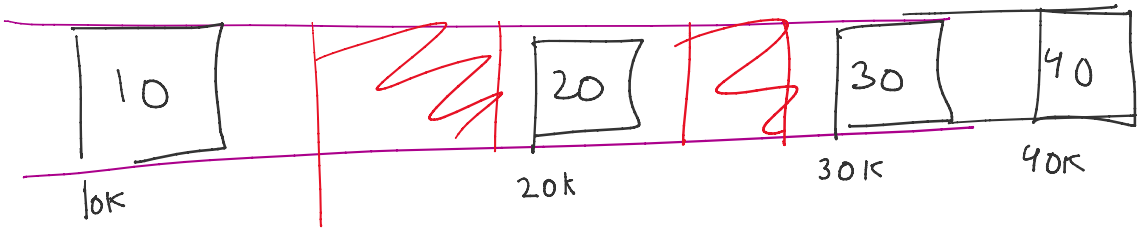
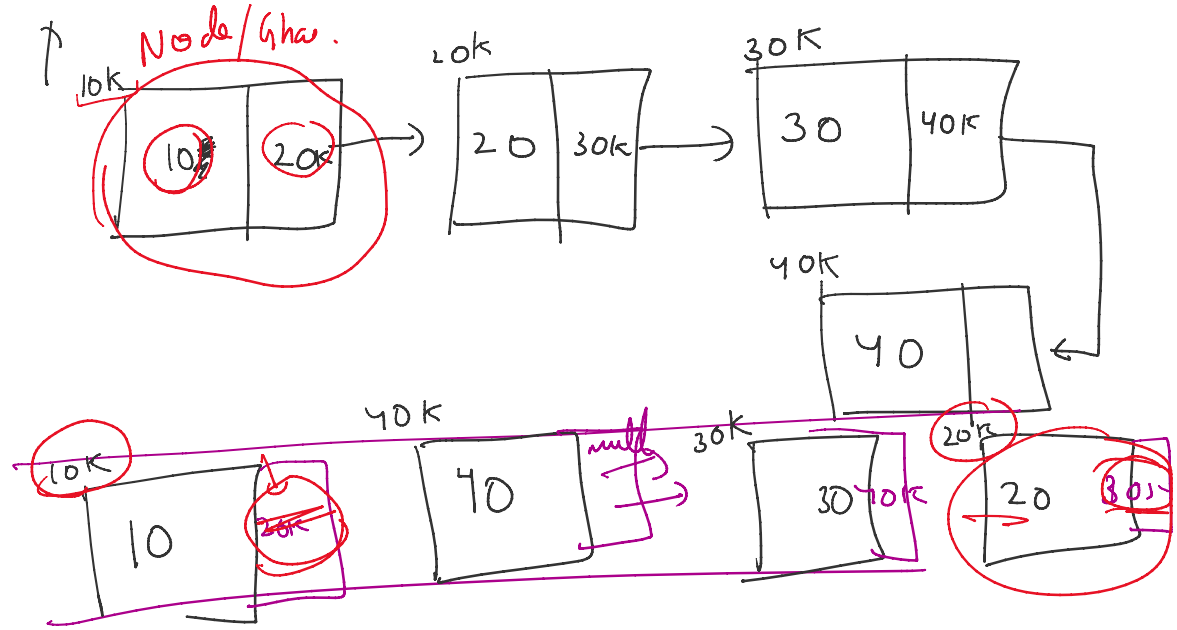
25 June 2023 10:21



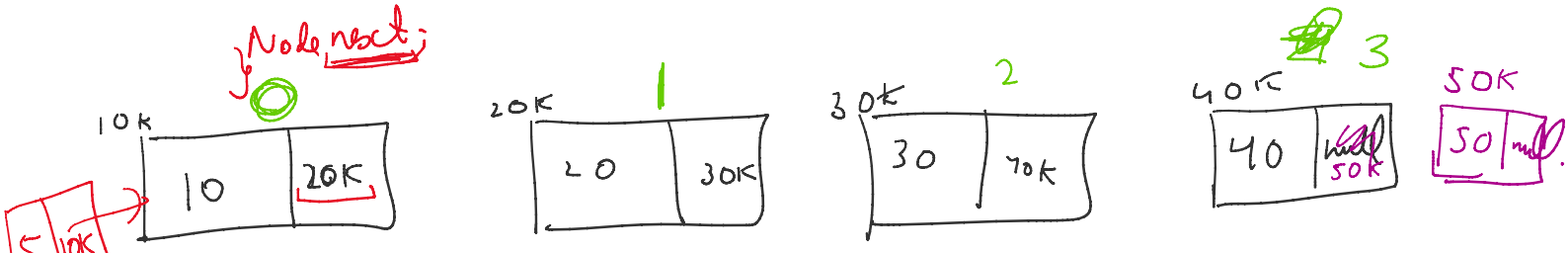









```
Node {
    int data
    Node next;
}
```



temp = 10K

Syso(temp.data)

temp = temp.next

10K.next = 20K

Syso(20K.data)

temp = 20K.next = 30K

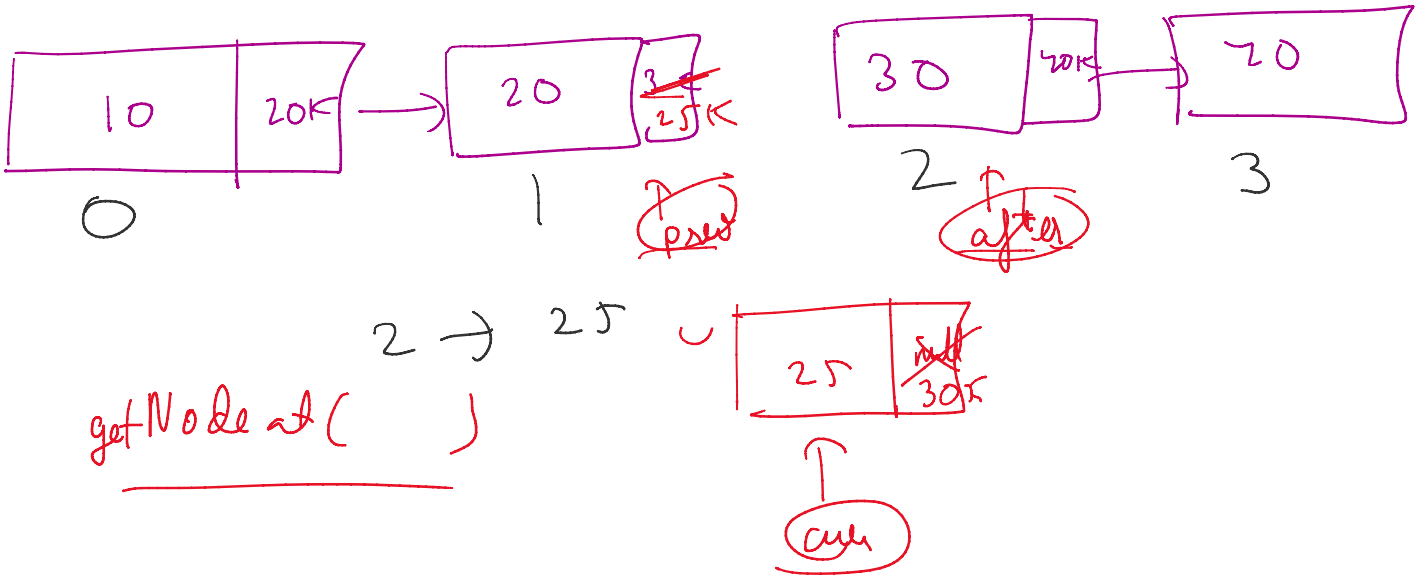
Syso(30K.data)

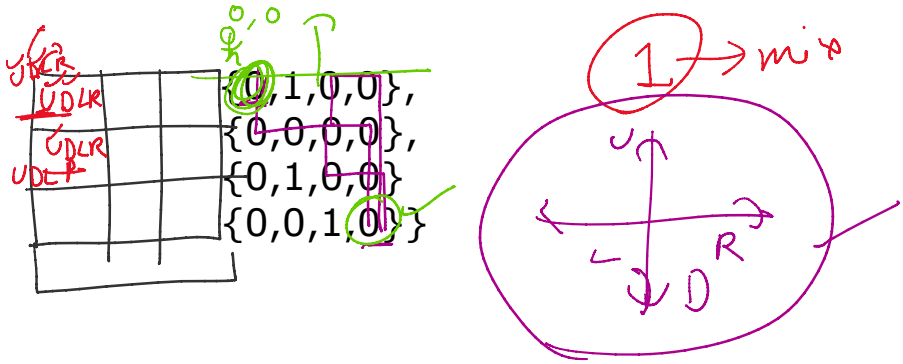
temp = 30K.next = 70K

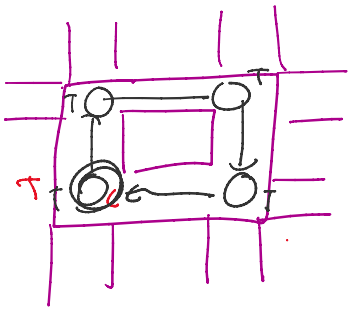
Syso(40K.data) → 0

- Display
- size.
- getFirst; first Node data (Incomplete)
- getLast; Last Node data how to find ?
  - Diff between size, getLast

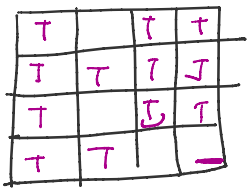
- size.
- getFirst; first Node data (Incomplete)
- getLast; Last Node data how to find ?
  - Diff between size, getLast
- getAt(idx);
  - Indexing starting from 0
  - Invalid when idx<0 and idx>=size()
- addLast(50)
  - Create Node then move to last(assuming u can) Node Explain Linking
- isEmpty ()
- Now edit addLast(),getLast,getFirst;
- addFirst();
- addAt(indx);

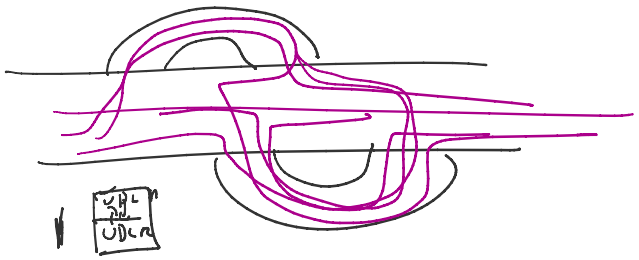




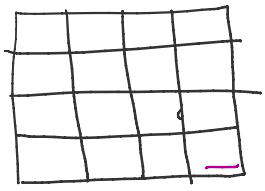


{0,1,0,0},
{0,0,0,0},
{0,1,0,0}
{0,0,1,0}}





{0,1,0,0},
{0,0,0,0},
{0,1,0,0}
{0,0,1,0}}



B ADE

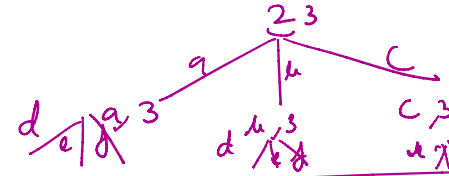
A	B	C	E
S	F	C	S
A	D	E	E

SEECS

Input: digits = "23"
Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]

From <https://leetcode.com/problems/letter-combinations-of-a-phone-number/>

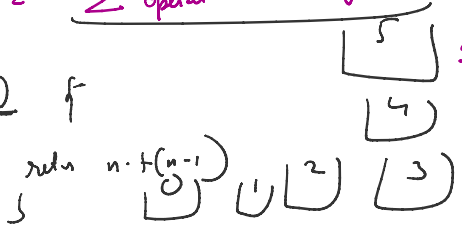
6 8 7 9
3x3 x4x4 = 36 ✗ =



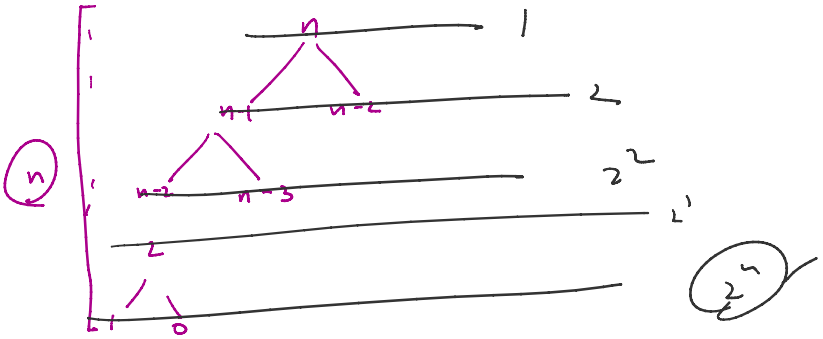
Santa

→ f(n) = Σ operat^n in each fn frame.
= (no. of f^n frame) x avg. work in each f^n frame

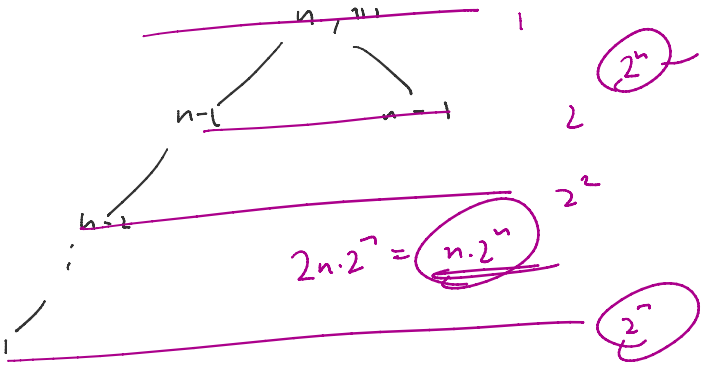
f(n)

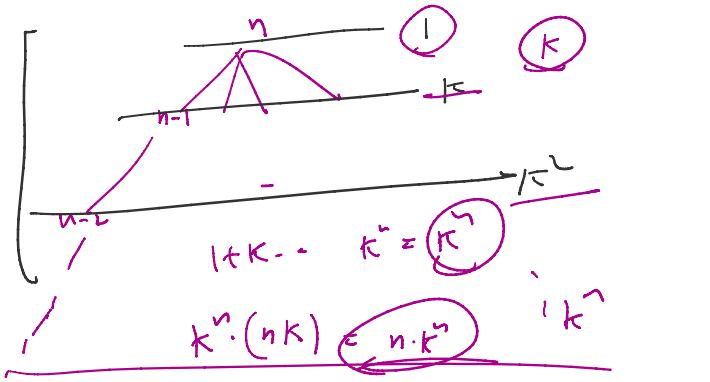


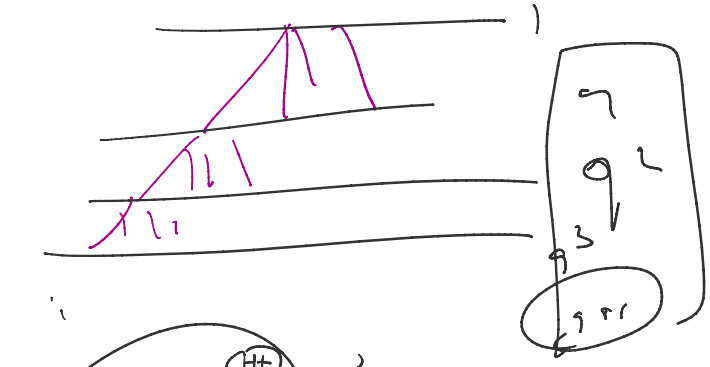

```
f(n) {
  ... f(n-1) + f(n-2)
}
```

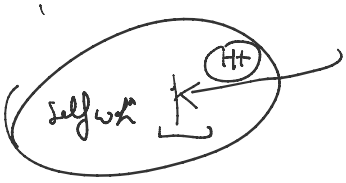


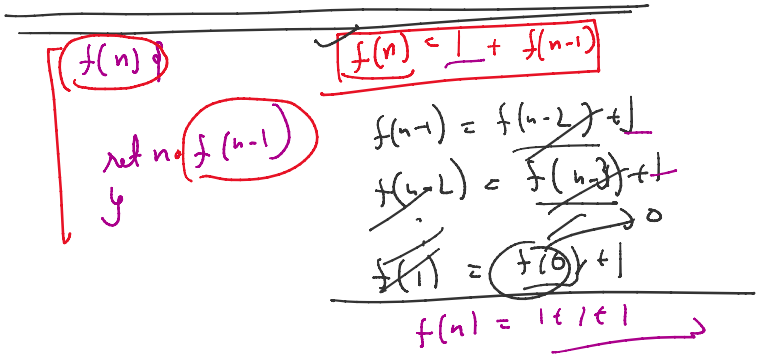
$2^0 + 2^1 + \dots 2^n = 2^{n+1}$











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

$f(n-1) \geq f(n-2)$

$f(n) \leq 2f(n-1) + 1$

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

$2f(n-1) = 2^2 f(n-2) + 1 \cdot 2$

$2^2 \cdot f(n-2) = 2^3 f(n-3) + 1 \cdot 2^2$

$2^3 f(n-3) = 2f(n-4) + 1 \cdot 2^3$

$2^{n-1} \cdot f(1) = 1 \cdot 2^{n-1}$

$f(n-(n-1))$

$f(n) = 2^n$

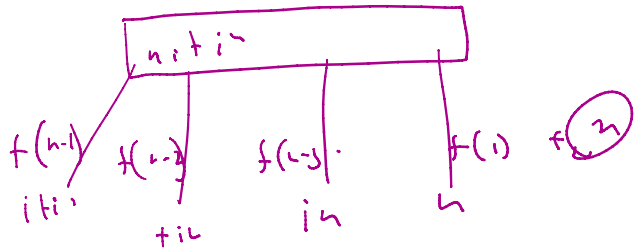
LKC

$f(n) = kf(n-1) + nk$

$kf(n-1) = k^2 f(n-2) + (n-1)k^2$

$k^n \cdot n$

$n \cdot k^n$



$f(n) = f(n-1) + f(n-2) + \cdots + f(1) + n$

$f(n-1) = f(n-2) + f(n-3) \cdots f(1) + (n-1)$

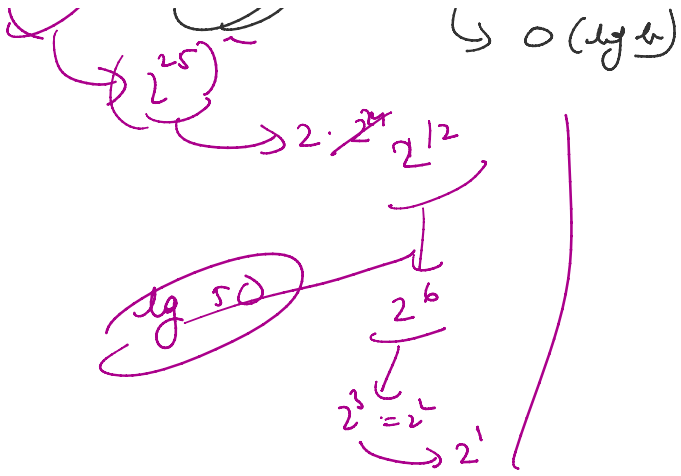
$f(n) = 2f(n-1) + C$

$2^{50} \rightarrow 50$

2^{25}

$a^n \rightarrow O(n)$

$\hookrightarrow O(\log n)$


$\log n$
2^{25}
$2 \cdot 2^{24}$
2^{12}
$\log 50$
2^6
$2^3 = 2^2$
2^1