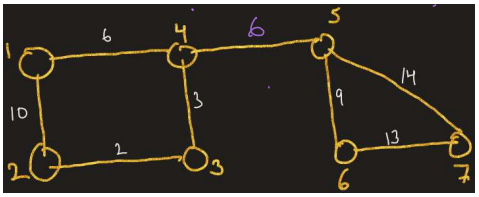
Sunday, August 28, 2022 10:43 AM

23456
234576

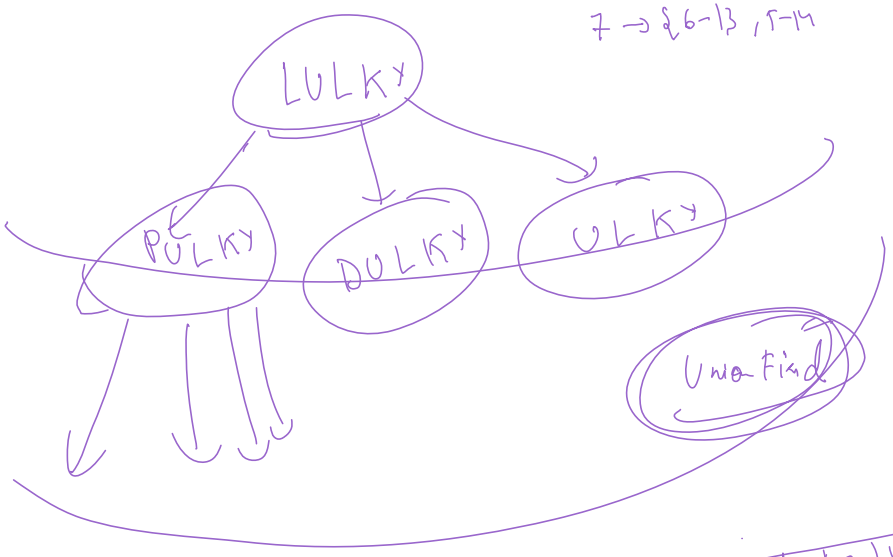


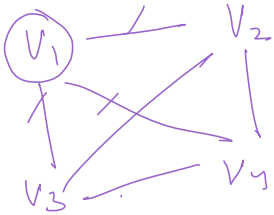
, u?

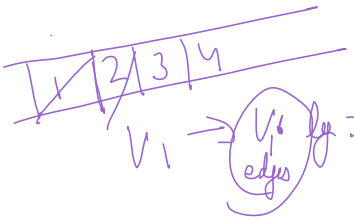
∠ V_2

1 → , -10 }
-1 , 3-2 }
, 4-3 }
-6, 3-3 ,

{ 6-9 , 7-14
→ { 5-9, 7-13,
7 → { 6-13 , 5-14





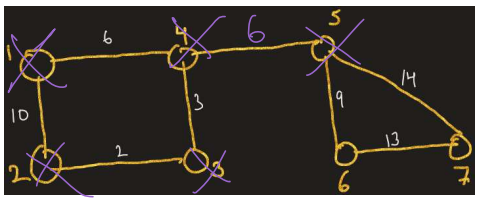


V_2 → V_{ges} · log x

E log x

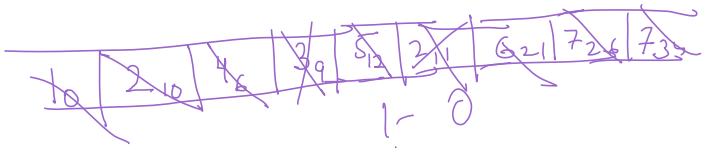
x → V^2

Dij → E log V



PQ <Dij P>

↳ V
↳ Wt



1- 0

4-6

3-9

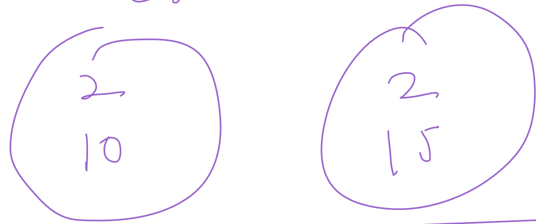
2-10
5-12

6-21

7-26

14
12
26

7-26



min cost path

0,2

1	3	1
1	5	1
4	2	1

x min

+ grid [r, c]

sp1 = (r+1, c)

sp2 = (r, c+1)

min (sp1, sp2) + grid [r] (c)

Tower of Hanoi

f^n (n → A to C ; B)

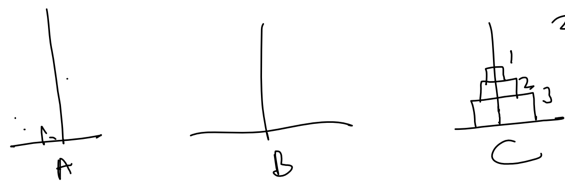
n, src, dst, H

n, src, H, dst

(nth disk, src, dst)

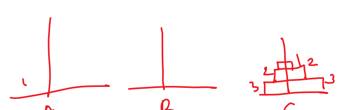
n-1, H, dst, src

1 → A to C
2 → A to B
1 → C to B
3 → A to C
1 → B to A



1 → A to C
2 → A to B
1 → C to B
3 → A to C

1 → B to A
2 → B to C



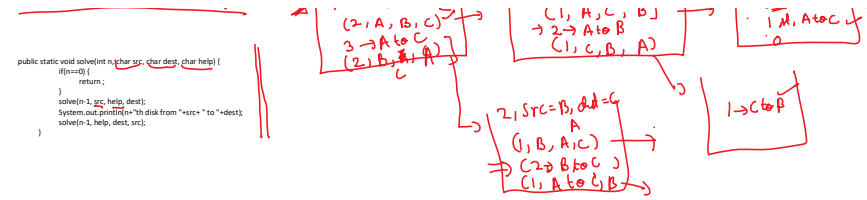
3, src=A, dest=C, B
(2, A, B, C)
3 → A to C
(2, B, A, C)

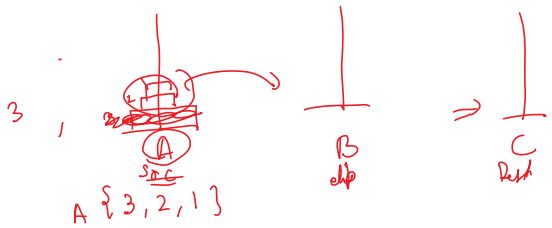
2, src=A, dst = B, C
(1, A, C, B)
→ 2 → A to B
(1, C, B, A)

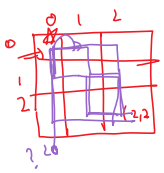
1, src A, dest=C
H = B
1 M, A to C

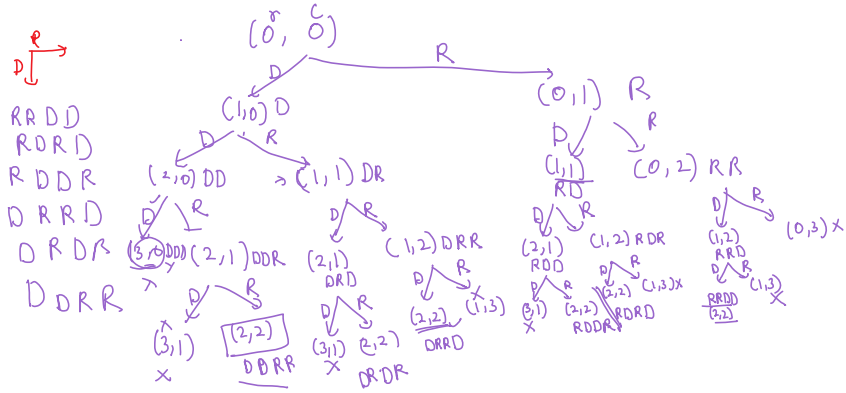
```
public static void solve(int n, char src, char dest, char help) {
            if(n==0) {
                  return ;
            }
            solve(n-1, src, help, dest);
```

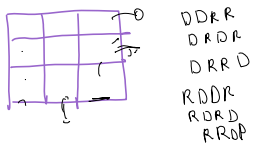
```
public static void solve(int n,char src, char dest, char help) {
        if(n==0) {
            return ;
        }
        solve(n-1, src, help, dest);
        System.out.println(n+"th disk from "+src+ " to "+dest);
        solve(n-1, help, dest, src);
    }
```

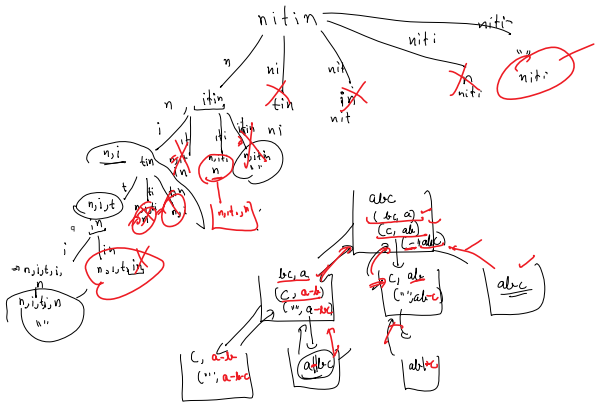


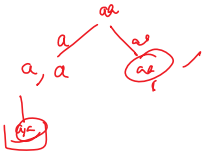


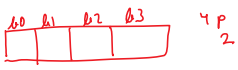


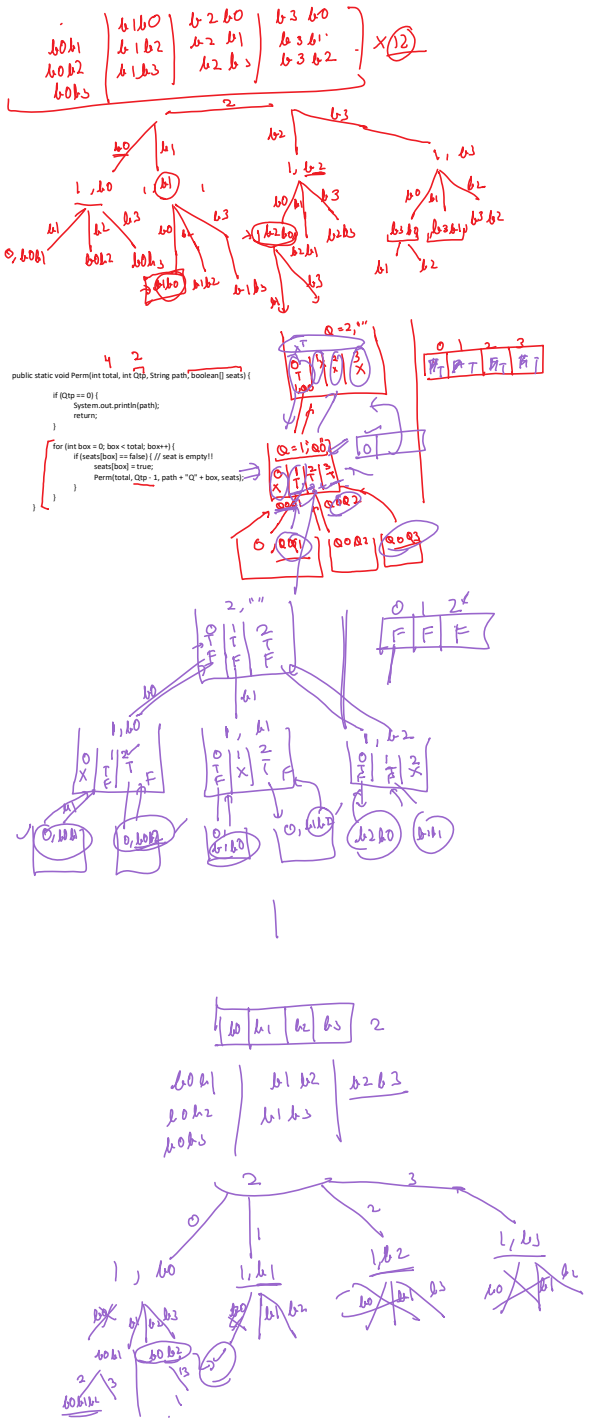











4 2
public static void Perm(int total, int Qtp, String path, boolean[] seats) {
if (Qtp == 0) {
System.out.println(path);
return;
}
for (int box = 0; box < total; box++) {
if (seats[box] == false) { // seat is empty!!
seats[box] = true;
Perm(total, Qtp - 1, path + "Q" + box, seats);
}
}
}