Recursion (Print)

07 January 2023 17:08

1) Coin Toss

n coins

H T

n=1 H , T → 2^1

n H , HT , TH , TT → 2^2

n=3 HHH TTT HHT TTH HTH THH HTT THT → 2^3

n=4 ⇒ (n=3 total solutions) * (H + T) ✓

1) B.P?? CT(n)

2) SP?? CT(n-1)

3) S.W H * CT(n-1) + T CT(n-1)

4) B.C 0 ✓

coinToss (n)

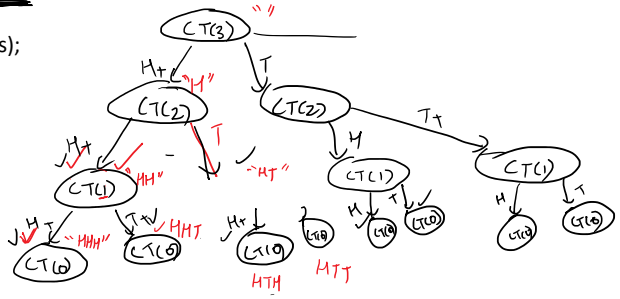
↳ H + coinToss(n-1)

T + coinToss(n-1)

```
public static void ct(int noc, String ans) {
    if (noc == 0) {
        System.out.println(ans);
        return;
    }
    ct(noc - 1, ans + "H");
    ct(noc - 1, ans + "T");
}
```

ct(1, HT) {
ct(0, HT + H);
ct(0, HT + T);
}

HHH HHT



HTH HTT THH THT TTH TTT

Subarray ?? Substring ??

↳ Contiguous subpart

Subset → NON CONTIGUOUS Subsequence →

↳ A: { 10, 20, 30, 40 }

B { 10, 30 } C: { 10, 40 }

B ⊂ A

s1: "abc"

s2: "ac" ✓

sb: "cb" ✗

"abc" → Print All possible subsequences

abcd

"abc" → Print All possible subsequences

"" "a" "b" "c"

"ab", "ac", "bc"

"abc"

↳ 2 → include
↳ exclude

✓ ✗ ✓ ✗ ✓ ✗

✓ ✓ ✓ "abc"

⑧ 2^3

a b c
2 2 2

✓ ✗ ✓ "ac"

✗ ✗ ✓ "c"

✓ ✓ ✗ "ab"

✗ ✗ ✗ ""

a b c d
$2 \times 2 \times 2 \times 2$

1) P.SS.("abc")

S.P. : P.SS.("bc") ✓

↳ "", "b", "c", "bc",
"ab", "ac", "abc"

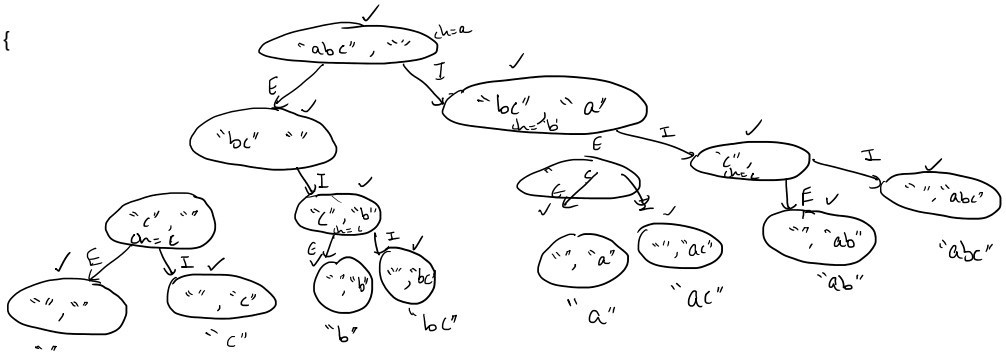
B.C. : PSS("") → return

a E → ""
I → "a"

```
public static void pss(String str, String ans) {
    if (str.length() == 0) {
        System.out.println(ans);
        return;
    }
    char ch = str.charAt(0);
    String rest = str.substring(1);

    // exclude ch
    pss(rest, ans);

    // include ch
    pss(rest, ans + ch);
}
```



Print All subsequences with ASCII

"b"

"", "b", 98
↓ ↳
Exclude Include Include ASCII

"ab"

"", "b", "98"

"a", "ab", "a98",

"97", "97b", "9798"

3 3 $= 3^2$

'a' → 1
'b' → 2

↳ ch - 'a' + 1

'c' 97 +1
= 3

Keypad Combination

245

1) B.P ⇒ K.C(245)

2) S.P ⇒ K.C(45)

B.C ??
↳

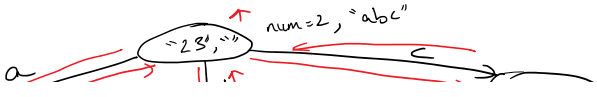
2 → abc
4 → ghi
5 → jkl

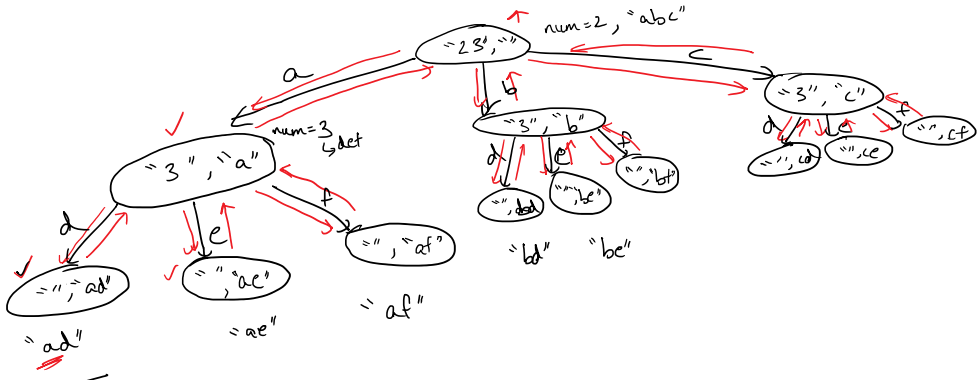
1) convert(2) → "abc"
convert(7) → "pqrs"

S.W → 2 ke + SP

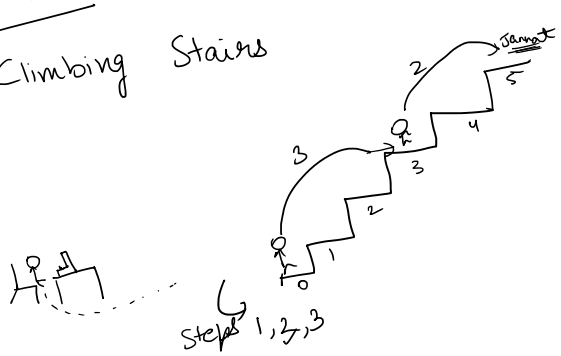
Combinations
a,
b,
c

('a' + {SP}),
('b' + {SP}),
('c' + {SP}),





Climbing Stairs



111111
1112
32
212
311
131
23
122

How many ways??

How many SP??

0 → 1, 2, 3

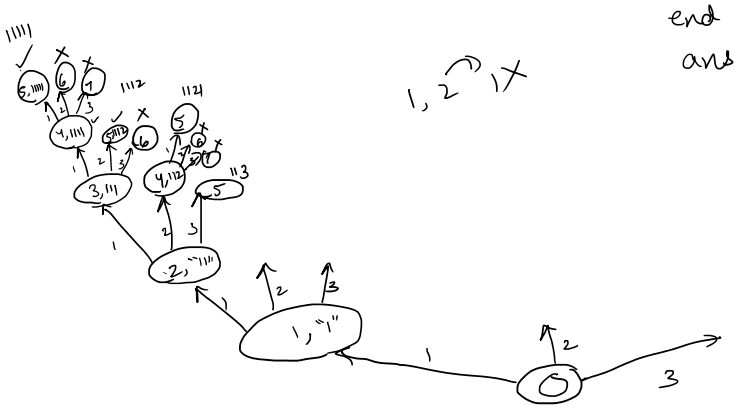
0, 1, 2, 3

Base Case??

↳ +ve B.C → Jannat ↳ curr = end

↳ -ve BC → Antriksh ↳ curr > end

current posn
end posn
ans



1, 2 ✓ X

Perm { 122, 212, 221 } → 1

Combinations { 113, 131, 311 } → 1

23, 32 → 1

1112 1211 } 1
1121 2111

111 2 ✓
0 1211
11 21
2111

Unique Jumps

Last Jump Kondi??

1112,

[23, 11111, 122, 2117]

n=1 ()✓)(✗

n=2 ()() (())

n=3 ()()() (()) (()()) ()(()) (())()

(((()) ✗

)()((✗

(()))✗)(()

))(✗

))✗

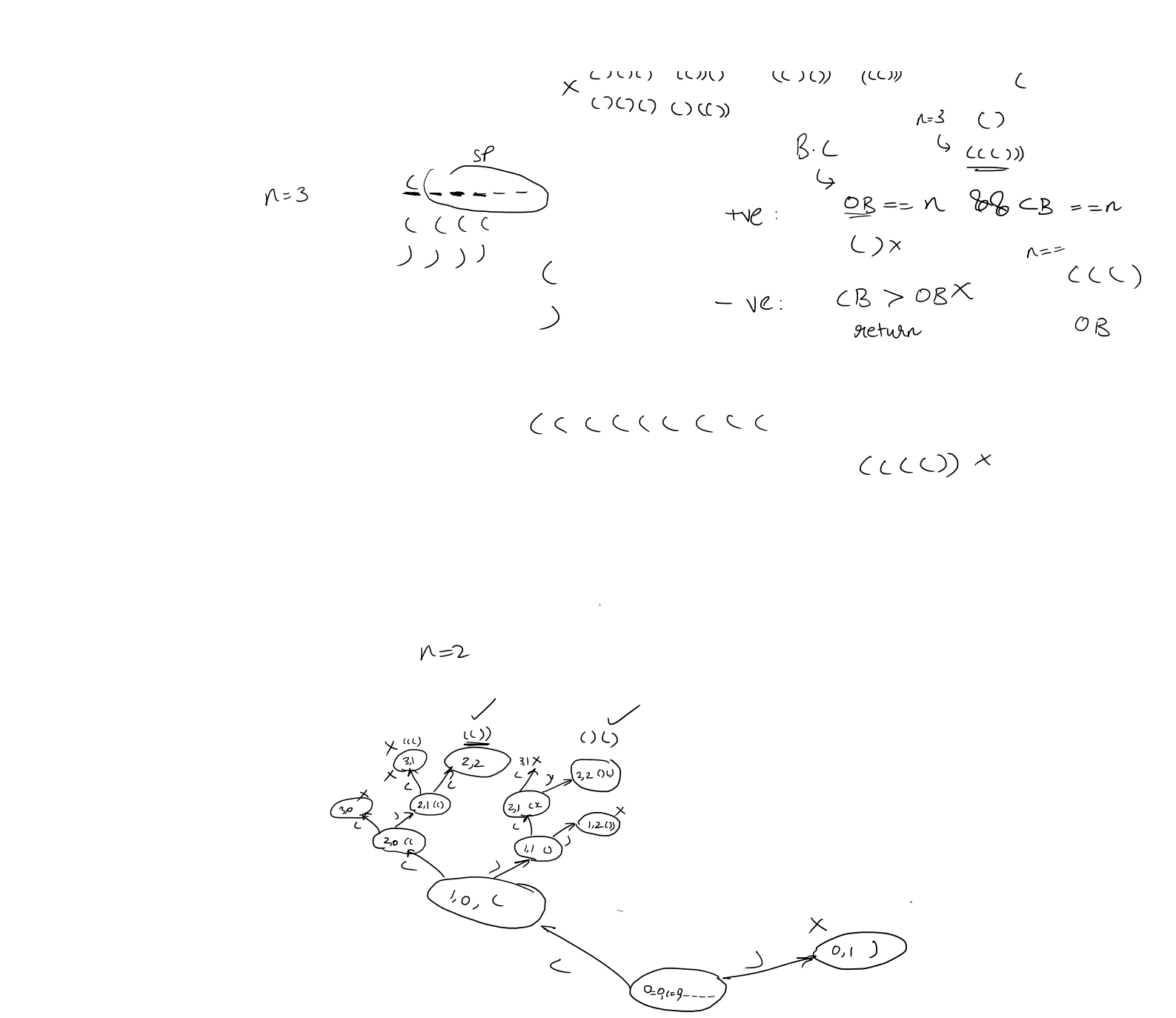
BP: GP(n)

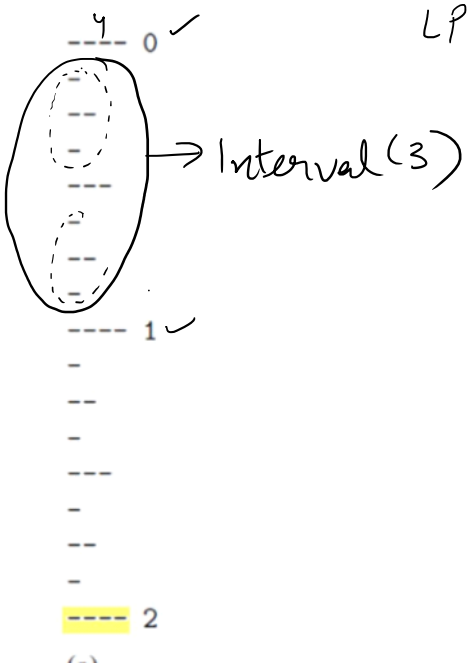
SP: GP(n-1)?? ✗

✗ ()
()() (())
()()() (())() (()()) (())()
()()() ()(())

n=3 ()

(


n=3
SP
+ve :
OB == n && CB == n
-ve :
CB > OB
return
OB
n=2
3,1
2,2
3,0
2,1
2,0
1,1
1,2
1,0
0,1
0,0



LP 0 ---- 0

---- 1

---- 2

B.P → Int(3)

SP → Int(2)