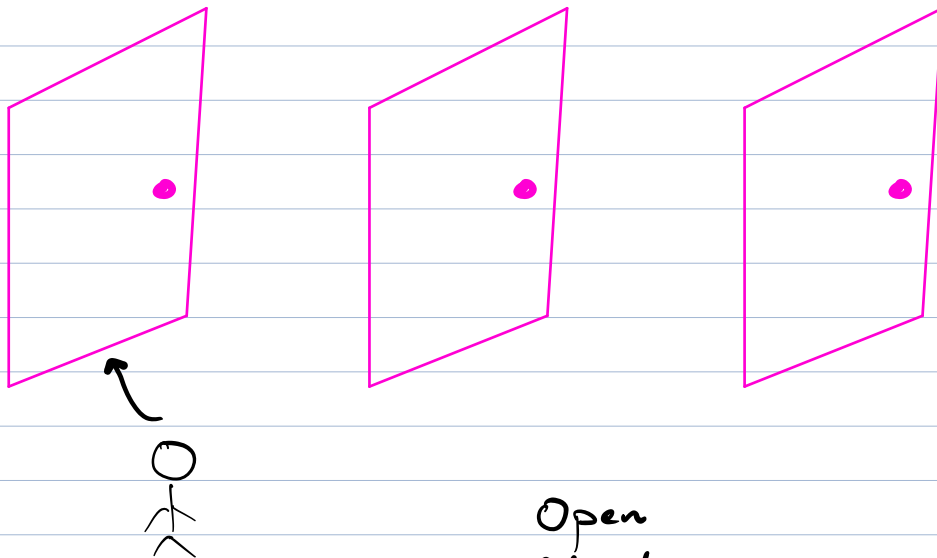


$$\text{Total dist} = 25 + 5 + 12 = 42$$

min distance from A to B = 42

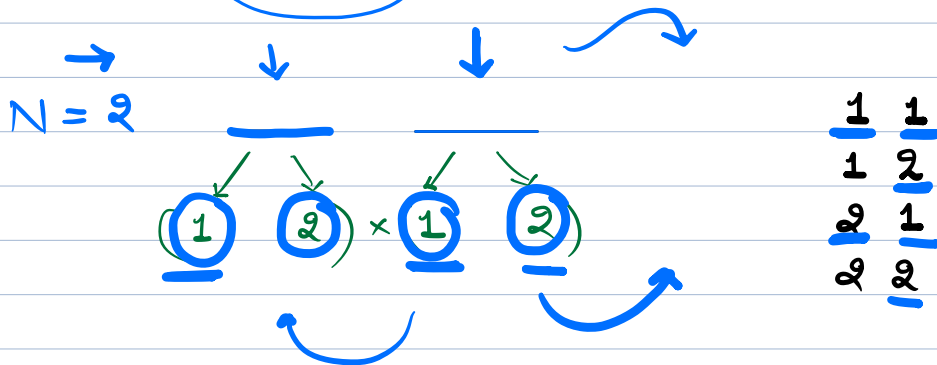
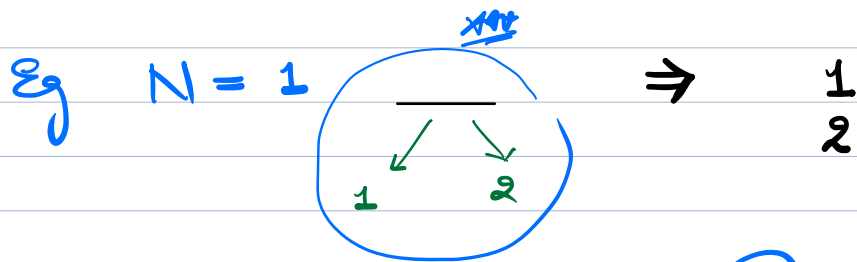
Ans



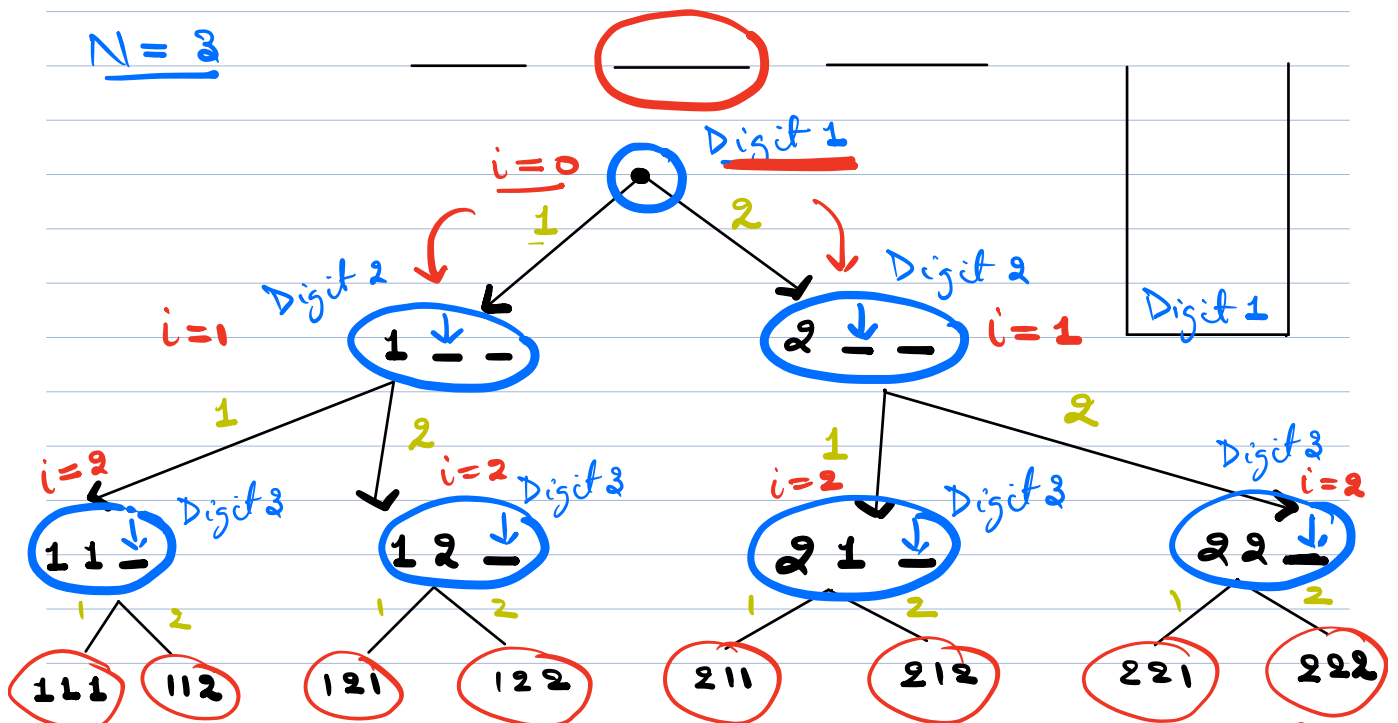
Open
check
close → Backtrack

Backtracking ⇒ Trying out all possibilities.

Print all N digit no's using the digits { 1 & 2 }



$N = 3$



Base Case ($i = 3$) ($i == N$)
Recursion \Rightarrow DFS ($[0, N-1]$)

Code

int currNumber[N], \downarrow

void generate (N, currNumber, index) \uparrow

\rightarrow if (index == N) {

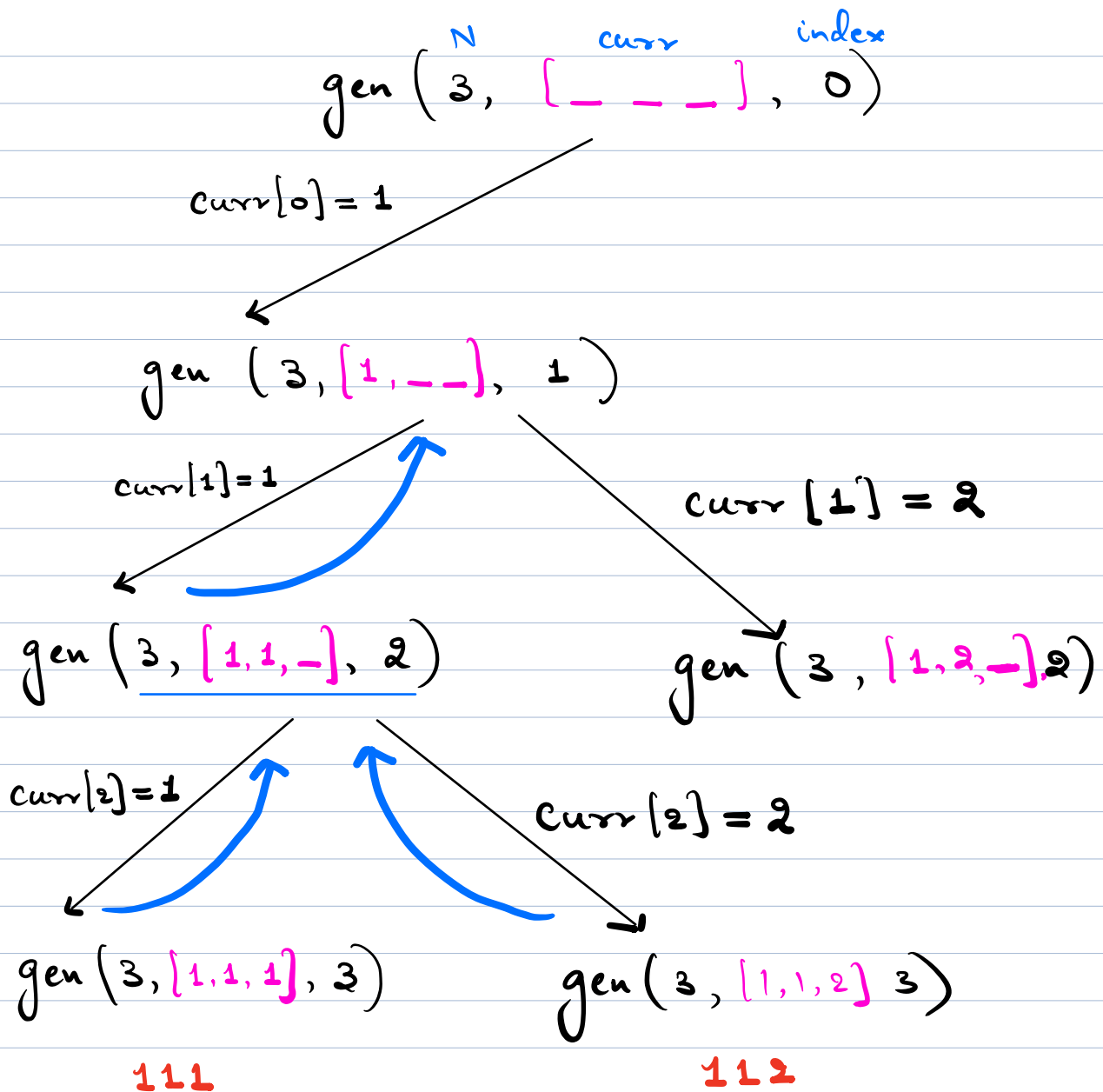
$O(N)$ print currNumber array;
return;
}

\rightarrow currNumber[index] = 1;

1st poss. generate (N, currNumber, index+1);

2nd poss. \downarrow
currNumber[index] = 2;
generate (N, currNumber, index+1);

}



T.C. =

$$\left[\left(2^0 + 2^1 + 2^2 + \dots + 2^{N-1} \right) \times 1 + \left(2^N \right) \times N \right]$$

$$= (2^N - 1) + N \times 2^N$$

$$= 2^N (N+1) \quad \text{---1}$$

$$\text{T.C.} = \underline{N \times 2^N}$$

$$\text{S.C.} = \underline{\underline{O(N)}}$$

Q Print all N digit no's using
 { 1, 2, 3, 4 & 5 }

```
void generate (N, curr, index) {
```

Base Case

```
    currNumber[index] = 1;
    generate (N, curr, index+1);
```

```
    currNumber[index] = 2;
    generate (N, curr, index+1);
```

```
    currNumber[index] = 3;
    generate (N, curr, index+1);
```

```
    currNumber[index] = 4;
    generate (N, curr, index+1);
```

```
    currNumber[index] = 5;
    generate (N, curr, index+1);
```

}

```
for (i = 1; i <= 5; i++) {
    currNumber[index] = i;
    generate (N, curr, index+1);
}
```

$$T.C. = O(N \times 5^N)$$

~~Q~~

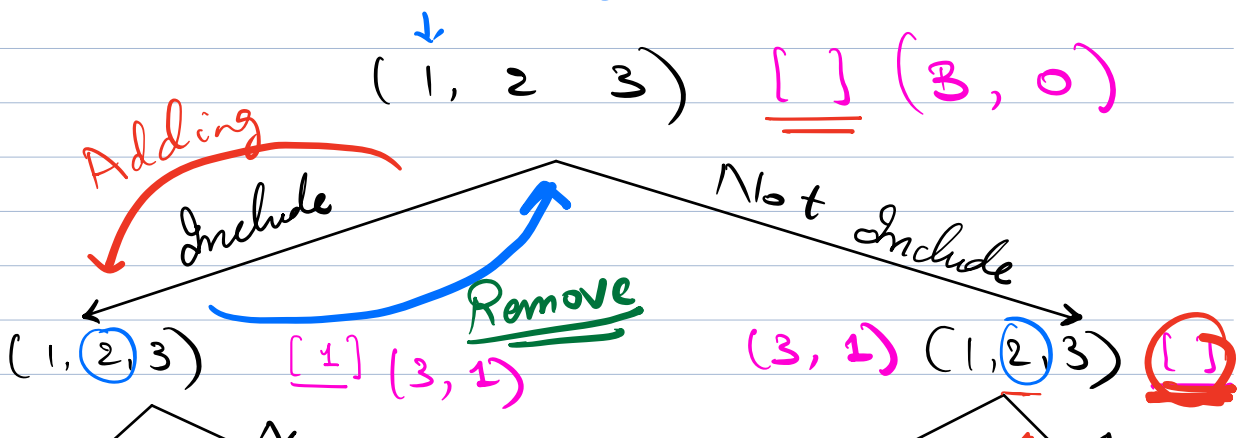
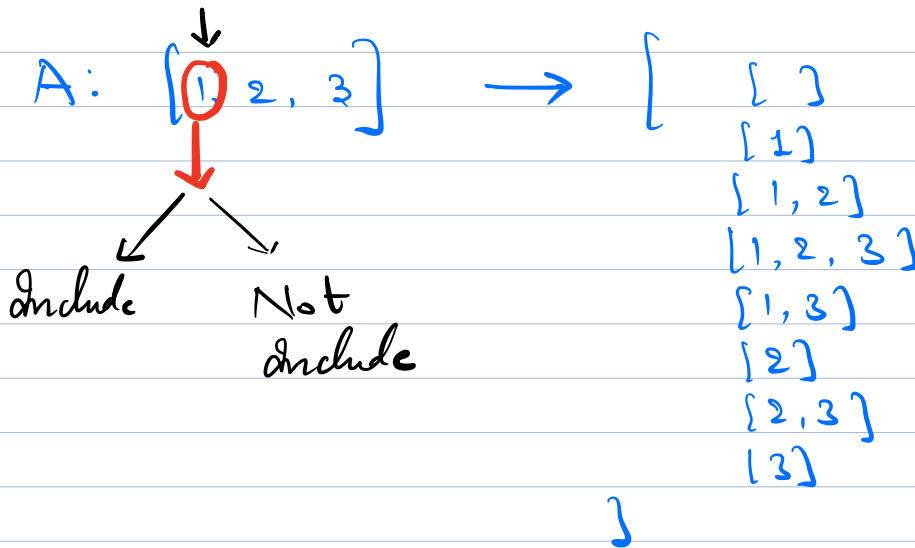


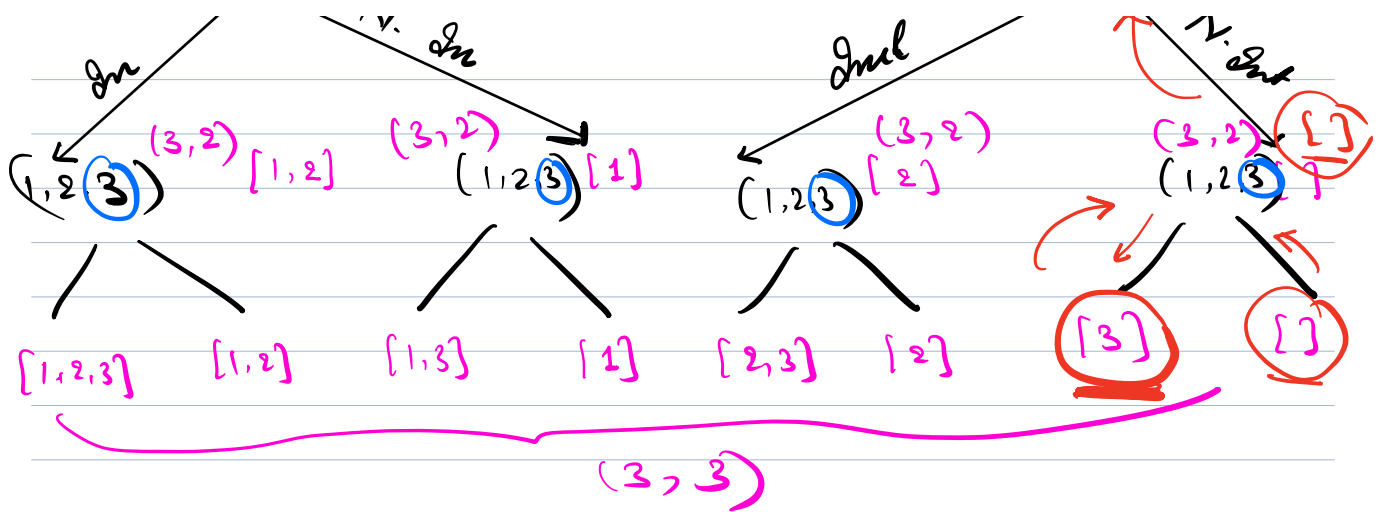
What is the choice for every element.

Q

Given an array of size N

Generate all subsets of the array





```
void printAllSubsets (N, set<Integer> ans, index) {
    if (index == N) {
        print ans set;
        return;
    }

```

```
    ans.add (A[index]);
    printAllSubsets (N, ans, index+1);

```

```
Undo → ans.remove (A[index]);
    printAllSubsets (N, ans, index+1);

```

```
    }

```

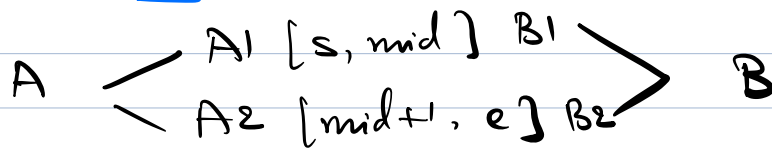
$$T.C. = O(N \times 2^N)$$

H.W.

Count the no. of subsets
with sum = (k)
input

$$\text{T.C.} = \underline{\underline{O(2^N)}}$$

merge (A ^B s, mid, e) &
sort



index = 0 , i = 0, j = 0
A1 A2

if (A1[i] <= A2[j]) &

A[index] = A1[i];
B[index] = A2[j];
index++;
i++;

;

Toggle

2^8

G

$$(N - (2^H - 1)) , \text{ (Diagram)}$$

