

Recursion

Lecture - 28

Page No.:

Date: / /

* House Robber : (Leetcode) :

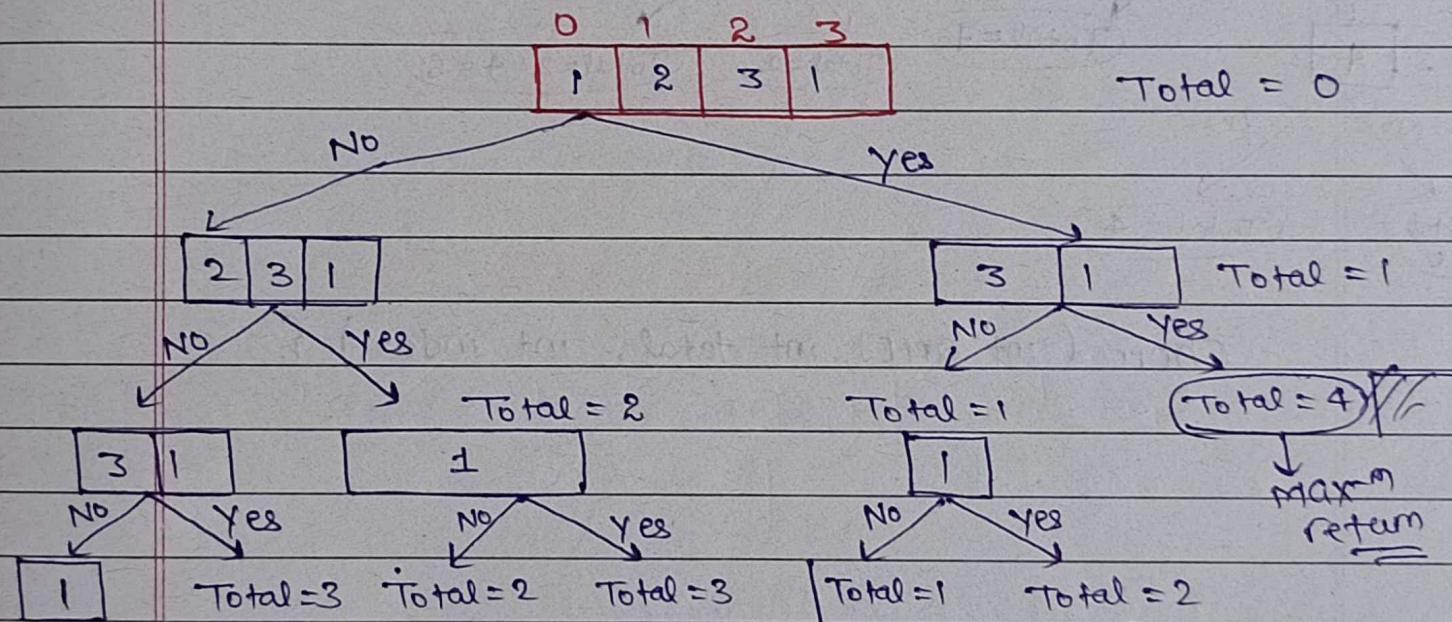
nums = [1, 2, 3, 1]

योरी Alternate दूर से करना है।

House	0	1	2	3
Money	[1]	2	3	1

[] [] []

either (0 & 2) or (1 & 3)



House : 0 1 2 3 4

Money : 1 2 3 1 4

Alternate graph में योरी बढ़ागा

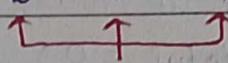
Maximum kitna योरी अब बढ़ता है।

↓

return.

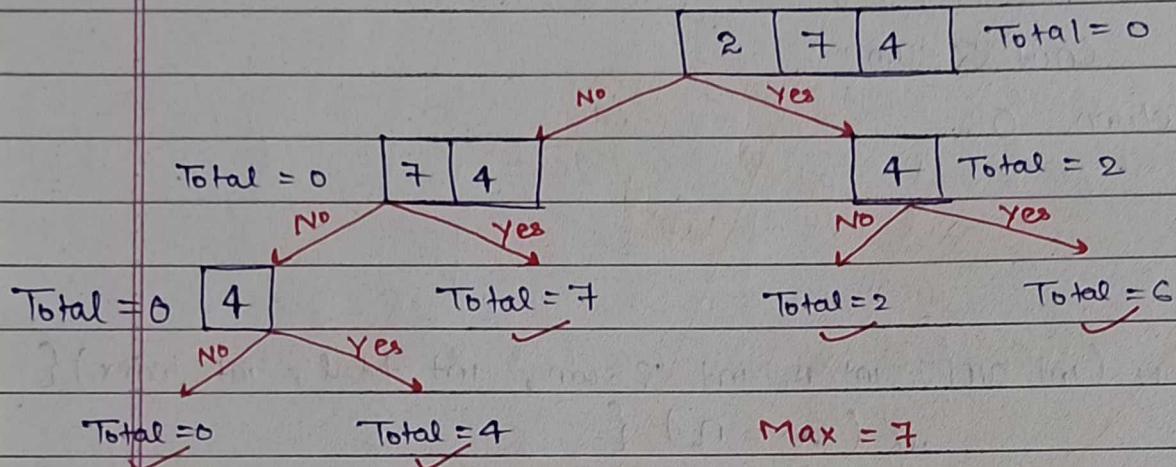
House 0 1 2

Money 2 7 4



$$2+4=6$$

(7) \rightarrow Max



Code in C++ :

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int arr[] = {2, 7, 4};
```

```
    int total = 0;
```

```
    int index = 0;
```

```
    int sum = 0;
```

```
    chori(arr, 3, sum, total, index);
```

```
    cout << sum;
```

```
    return 0;
```

}

```
int main() {
    int n;
    int arr[n];
    cin >> n;
    int sum = 0;
    int ans = Chori(arr, n, sum, total, 0);
}
```

```
Void Chori (int arr[], int size, int &sum, int total,
            int index) {
```

```
    if (index >= size) {
        sum = max (sum, total);
        return;
    }
}
```

```
Chori (arr, size, sum, total, index + 1);
```

```
Chori (arr, size, sum, total + arr[index], index + 2);
```

```
}
```

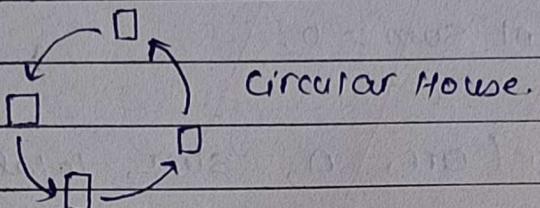
This Codes gives Time limit Exceed.

*

House Robber - II

Circular array

Alternate House से आर्टी ।



House	0	1	2	3	X
Money	2	4	3	7	

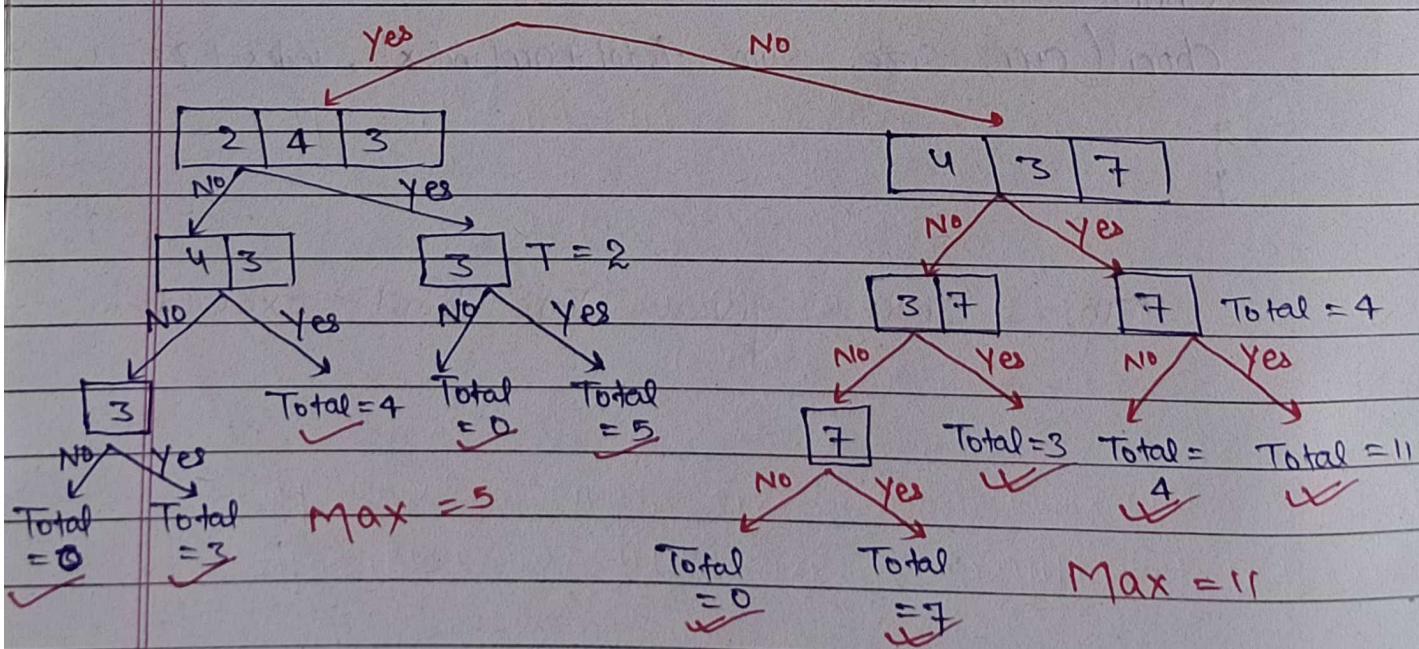
{2} {4} {3} {7}

{2, 3} {4, 7} {x}

↳ Maximum = 11

↳ return.

0	1	2	3
2	4	3	7



*

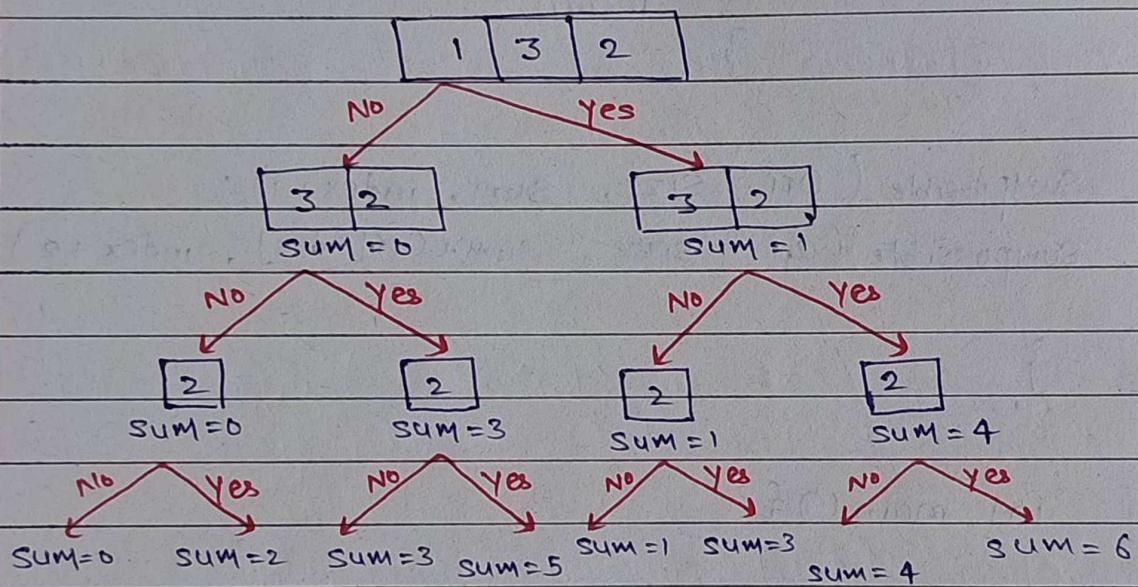
Combination Sum :-

2 1 4

{ } {2} {1} {4}

{2+1=3} {1+4=5} {2+4=6}

{2+1+4=7}

8 possibility

Sumpossible(int arr[], size, sum, index) {

if (index >= size) {

cout << sum << " ";

return ;

}

sumpossible(int arr[], size, sum, index + 1);

sumpossible(int arr[], size, sum + arr[index], index + 1);

}

→ check that sum = 7 is possible by any operation.

```

int main() {
    int arr[5] = { 2, 4, 1, 5, 8 };
    int index = 0, sum = 0;
    int ans = 0;
    int target = 7;
    ans = sumpossible( arr, 5, index, sum, ans, target );
    cout << ans;
}

```

```

int main() {
    int arr[5] = { 2, 4, 1, 5, 8 };
    int index = 0, sum = 0;
    sumpossible( arr, 5, sum, index );
    return 0;
}

```

House Robbery II (code)

```
Void find (vector <int> &nums, int i, int &sum, int total) {  
    if (i >= nums.size ()) {  
        sum = max (sum, total);  
        return;  
    };  
    find (nums, i + 2, sum, total + nums[i]);  
    find (nums, i + 1, sum, total);  
}
```

```
int robbery (vector <int> &nums) {  
    if (nums.size () == 1) return nums[0];  
    int sum = 0, total = 0;  
    find (nums, 1, sum, total);  
    total = 0;  
    nums.pop_back();  
    find (nums, 0, sum, total);  
  
    return sum;  
}
```

Check that the sum target is possible or not.

if possible return 1.
Not possible return 0.

only Minor change in code :

```
Void sumpossible (int arr[], int size, int index, int  
sum, int target, int &ans){
```

```
    if (index == size) {  
        if (sum == target) {  
            ans = 1;  
        }  
        return;  
    }
```

```
    sumpossible (arr, size, index + 1, sum, target, ans);
```

```
    sumpossible (arr, size, index + 1, sum + arr[index],  
                target, ans);
```

```
}
```

In Main :

```
ans = 0;
```

```
Cout << ans;
```

Code in Java :

```
public class TargetSum {  
    static boolean Sumpossible (int arr[], int size, int index,  
        int sum, int target) {  
        if (index == size) {  
            if (sum == target) {  
                return true;  
            }  
            else {  
                return false;  
            }  
        }  
        return Sumpossible (arr, size, index + 1, sum, target) ||  
            Sumpossible (arr, size, index + 1, sum + arr [index], target);  
    }  
}
```

```
public static void main (String [] args) {  
    int arr [] = {1, 2, 4, 7, 3};  
    int index = 0, sum = 0;  
    int target = 12;  
}
```

```
boolean ans = Sumpossible (arr, 5, index, sum, target);  
System.out.println (ans);  
}
```

```
}  
}
```

*

3, 2, 7

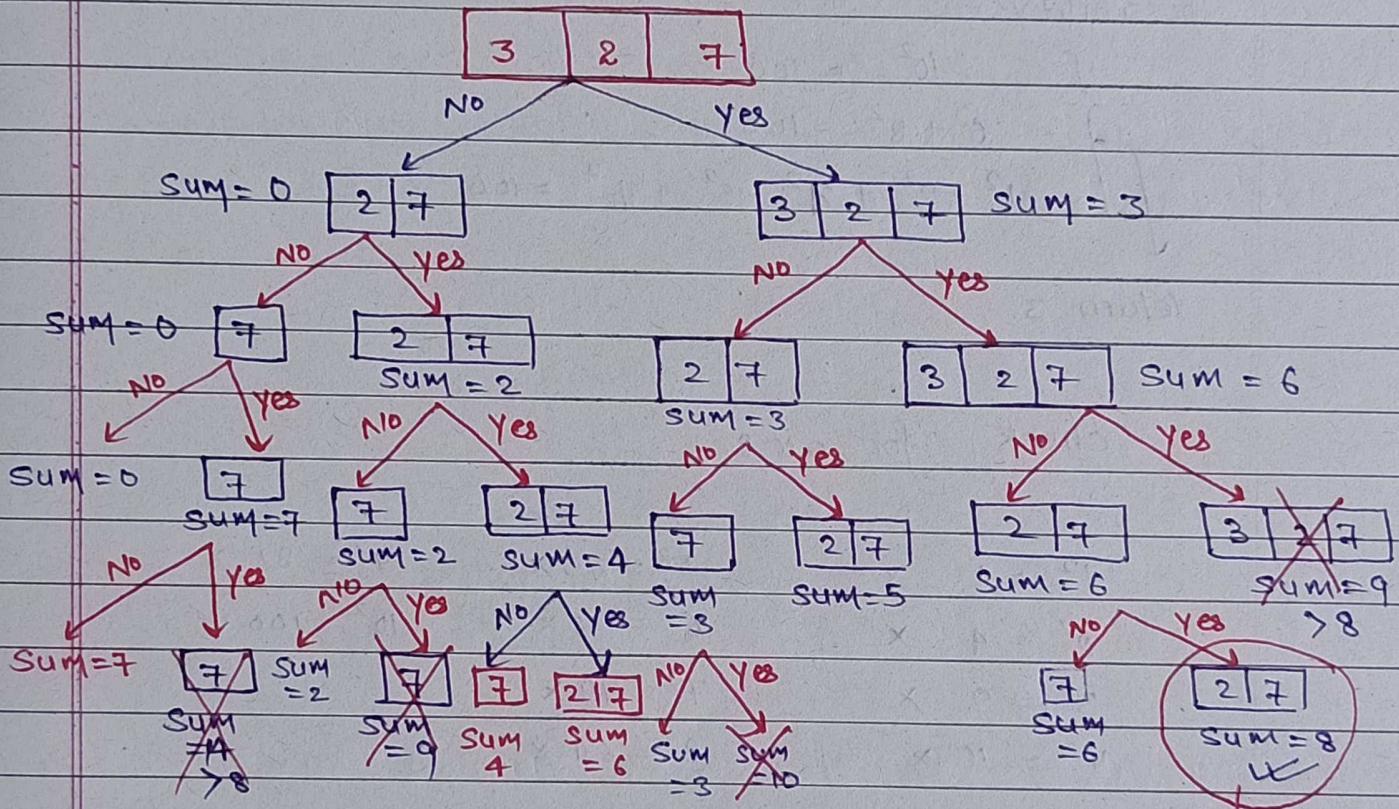
Target = 8

one number used many times.

Possibility :

$$3 + 3 + 2 = 8$$

$$2 + 2 + 2 + 2 = 8$$



Void sumpossible(int arr[], int size, int index, int total, int sum,

int &ans) {

if (index == size) {

if (sum == total) ans = 1;

return ;

}

if (sum > target) return ;

sumpossible(arr, size, index+1, total, sum, ans);

sumpossible(arr, size, index, total, sum+arr[index], ans);

* Count ways to express a number as
sum of powers

$$\text{Number} = 100$$

$$\text{Power} = 2$$

possibility ?

$$\left. \begin{array}{l} 10^2 = 100 \\ 6^2 + 8^2 = 100 \\ 1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 7^2 = 100 \end{array} \right\}$$

return 3.

check for all :

$$1^2 = 1 \times$$

$$5^2 = 25 \times$$

$$9^2 = 81 \times$$

$$2^2 = 4 \times$$

$$6^2 = 36 \times$$

$$10^2 = 100 \checkmark$$

$$3^2 = 9 \times$$

$$7^2 = 49 \times$$

$$4^2 = 16 \times$$

$$8^2 = 64 \times$$

$$\left\{ \begin{array}{l} \text{num} = 100 \\ \text{power} = 2 \\ \eta = 1, \text{Total} = 0 \end{array} \right\}$$

$$\begin{array}{c} \text{No} \quad \text{Yes} \\ \swarrow \quad \searrow \\ \text{num} = 100 \\ \text{power} = 2 \\ \eta = 2, \text{Total} = 0 \end{array}$$

$$\text{num} = 100$$

$$\text{pow} = 2$$

$$\eta = 10, \text{Total} = 0$$

$$\begin{array}{c} \text{No} \quad \text{Yes} \\ \swarrow \quad \searrow \\ \eta = 100 \text{ found.} \end{array}$$

Teacher's Signature.....

Sumpossible (int n, int num, int power, int total, int count){

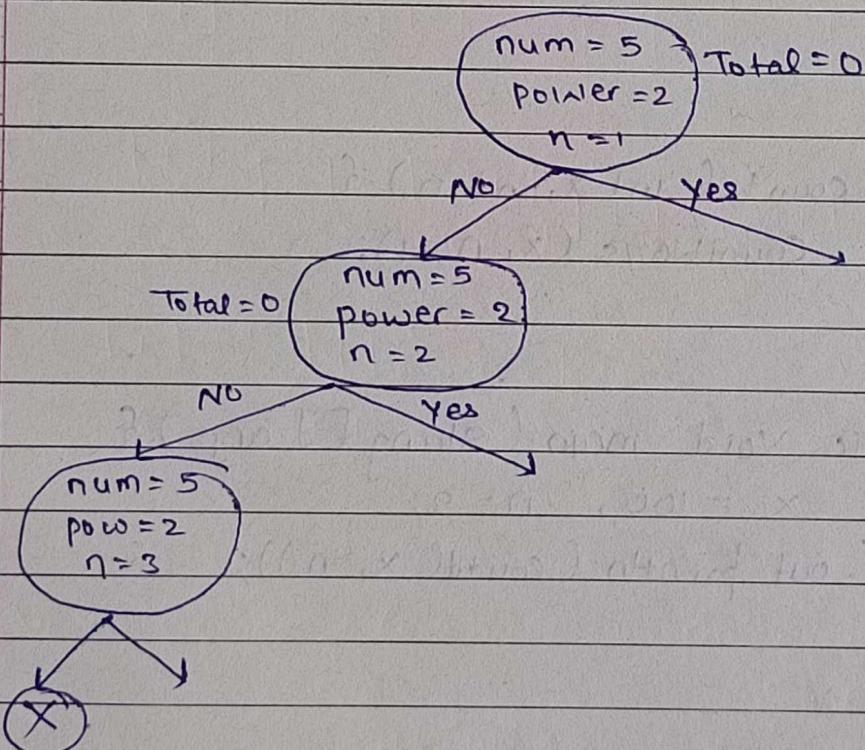
```
if (total == num) {
    Count++;
    return;
}
```

```
if (total > num)
    return n;
```

Sumpossible (n+1, num, power, total, count);

Sumpossible (n+1, num * power, total + pow(n, power), count);

}



Code (Java):

```
static int CountWays (int x, int n, int num) {  
    int val = (int)(x - Math.pow (num, n));  
    if (val == 0)  
        return 1;  
    if (val < 0)  
        return 0;  
  
    return CountWays (val, n, num + 1) +  
        CountWays (x, n, num + 1);  
}  
  
static int Count (int x, int n) {  
    return CountWays (x, n, 1);  
}
```

```
public static void main (String [] args) {  
    int x = 100, n = 2;  
    System.out.println (Count (x, n));  
}
```