

Practice Problems

Relevel
by Unacademy

Problem - 1

You have to return the total number of steps to make a number(n) to 0. You can perform the following operations.

1. Choose one digit from that number
2. subtract that chosen digit from the given number
3. Repeat this till the number becomes 0.

I/P: 27

O/P: 5

Problem - 1

Solution: <https://jsfiddle.net/gtgstg/o5ar6nus/376/>

```
const nullify = (N) => {  
  let table = new Array(N + 1).fill(1e9);  
  table[0] = 0;  
  for (let i = 0; i <= N; i++) {  
    let str = i.toString();  
    for (let c = 0; c < str.length; c++){  
      let xo = i - parseInt(str[c]);  
      table[i] = Math.min(table[i], table[xo] + 1);}  
    }  
  return table[N]; }  
let N = 27;  
console.log(nullify(N));
```

Problem - 2

You have to return the total number of steps to make a number(n) to 1. Perform these tasks to achieve the goal.

1. $n = n - 1$
2. if($n \% 2 == 0$) $n = n / 2$
3. if($n \% 3 == 0$) $n = n / 3$

I/P: 10

O/P: 3

Problem - 2

Solution: <https://jsfiddle.net/gtgstg/o5ar6nus/380/>

```
const solve = (table, i) => {  
  let mini = table[i - 1];  
  if (i % 2 == 0) {  
    mini = Math.min(mini, table[i / 2]);  
  }  
  if (i % 3 == 0) {  
    mini = Math.min(mini, table[i / 3]);  
  }  
  table[i] = mini + 1;  
}
```

Problem - 2

```
const unify = (N) => {  
  let table = new Array(N + 1).fill(0);  
  table[1] = 0;  
  for (let i = 2; i <= N; i++) {  
    solve(table, i)  
  }  
  return table[N];  
}  
console.log(unify(10));
```

Problem - 3

Alice once asked bob that how efficient are you in backtracking. Bob want to impress Alice so, he replied: I am the creator of the backtracking algorithm. Whatever situation you can give me I will be able to backtrack it. Now Alice asked him to print all the possible combinations in a given array arr, such that each combination has exactly the same sum i.e Sum. Now you have to help bob because valentine is near.

I/P: arr = [1,2,1] , sum = 2

O/P: [1,1], [2]

Problem - 3

Solution: <https://jsfiddle.net/gtgstg/o5ar6nus/385/>

```
const util = (index, t, arr, answer, tempo, i) =>{
  tempo.push( arr[i] );
  solve( i+1, t-arr[i], arr, answer, tempo );
  tempo.pop();
}
const solve = (index, t, arr, answer, tempo ) =>{
  if( t == 0 ){
    answer.push( Array.from(tempo) );
    return;
  }
}
```


Problem - 3

```
for( let i=index; i<arr.length; i++ ){  
  if( i>index && arr[i]==arr[i-1] ) continue;  
  if( arr[i] > t ) break;  
  util(index, t, arr, answer, tempo, i);  
}  
}  
const saveBob = (arr, sum) => {  
  let ans = [];  
  arr.sort();  
  solve( 0, sum, arr, ans, [] );  
  return ans;  
}  
let arr = [1,2,1];  
let sum = 2;  
console.log(saveBob(arr, sum));
```

Problem - 4

You got access to the Indian currency printing press. There are multiple machines that print some kinds of notes like(10, 20 50, etc). You can print as many notes as you want with those machines. Now, you are a very honest person so, you will not steal any money from there. But in India, there is a crisis of notes because other countries have burned your currency. Now you have to print the notes to get a sum of money X. And with the least amount of notes.

For example, if the notes are of{10,500,2000} and the desired sum is 1010, an optimal solution is 500+500+10 which requires 3 notes.

Problem - 4

Solution: <https://jsfiddle.net/gtgstg/o5ar6nus/391/>

```
const solve=(i, table, notes,j) =>{
  if (notes[j] <= i)
  {
    let res = table[i - notes[j]];
    if (res !== Number.MAX_VALUE && res + 1 < table[i])
    table[i] = res + 1;
  }
}

const util = (i, table, notes) =>{
  for (let j = 0; j < notes.length; j++)
  solve(i, table, notes,j)
}
```

Problem - 4

```
const minNotes = (notes, sum) => {  
  let table = new Array(sum + 1).fill(0);  
  table[0] = 0;  
  for (let i = 1; i <= sum; i++)  
    table[i] = Number.MAX_VALUE;  
  for (let i = 1; i <= sum; i++){  
    util(i, table, notes);  
  }  
  if(table[sum] == Number.MAX_VALUE)  
    return -1;  
  return table[sum];  
}  
  
const notes = [1,5,7];  
const sum = 11;  
console.log(minNotes(notes, sum));
```

Thank You!