How to Solve Recursion Problems

Created	@July 6, 2022 8:02 PM
Class	
Type	
Materials	
Reviewed	□

Recursion based solutions are generally very easy to write. They Just require a bit of imagination and assumptions to work out.

Recursion is closely related to PMI (Principal of mathematical induction).

Prove that sum of first N natural numbers is (N(N+1))/2

We try to break our problem into 3 parts.

Base Case: The smallest value of N for which we already know the answer, verify if the formula works for it.

For N = 1, we know that the formula works.

Assumption: We assume that for some N = k, the formula works

So if the formula works for N = k, then we can say that sum of first k natural numbers is (k(k+1))/2

Self work: We manually prove that if formula works for N = k acc. to our assumption then it will also work for k+1

Sum of first k + 1 natural numbers acc to formula: ((k+1)(k+2))/2

We already know that sum of first k natural numbers is (k(k+1))/2 and to calculate the sum of first k + 1 natural numbers, we can say that it will be equal to: sum of first k natural numbers + (k+1) ⇒ (k(k+1))/2 + (k+1) ⇒ ((k)(k+1) + 2(k+1))/2 ⇒ ((k+1)(k+2))/2

Hence proved

NOTE: The above three parts apply same to same for solving a recursive problem.

Factorial

Let’s have a function f(n) which can give us n!

How to implement f(n) ?

Base Case: What is the smallest value of n for which we already know the answer ?

For n = 1, f(n) = 1

Assumption: We assume that for some value n-1, function f works correctly and will give us f(n-1) i.e. (n-1)!

Self work: If we have the correct value of f(n-1) then to calculate f(n) we can just multiply n with f(n-1)

f(n) = n*f(n-1) ⇒ Recurrence relation

```
function f(n) {
  // n!
  if(n == 1) return 1; // base case
  let assume = f(n-1); // assumption
  let ans = n*assume;
  return ans;
}
```

Dice Combinations

CSES - Dice Combinations Your task is to count the number of ways to construct sum n by throwing a dice one or more times. Each throw produces an outcome between 1 and 6. For

https://cses.fi/problemset/task/1633

```
function f(n) {
  if(n == 0) return 1;
  let ans = 0;
  for(let k = 1; k <= 6; k++) {
    if(n-k < 0) continue;
    ans += f(n-k);
  }
  return ans;
}
```

Removing Digits

CSES - Removing Digits You are given an integer n. On each step, you may subtract one of the digits from the number. How many steps are required to make the number equal to 0? Input

https://cses.fi/problemset/task/1637

```
function getDigits(n) {
  const result = [];
  while(n > 0) {
    let digit = n%10;
    result.push(digit);
    n = Math.floor(n / 10);
  }
  return result;
}
function f(n) {
  if(n == 0) return 0;
  if(n < 10) return 1;
  const digits = getDigits(n);
  let ans = Infinity;
  for(let i = 0; i < digits.length; i++) {
    if(digits[i] == 0) continue;
    ans = Math.min(ans, f(n - digits[i]));
  }
  return 1 + ans;
}
```

Given a grid of mxn dimensions, where you are starting from the top left, and from any cell we can only go to either the cell on right or the cell on left. Find the total no of ways to reach the bottom right cell of the grid.