**Recursion**

The process in which a function calls itself directly.

Why we use Recursion?

because there are many problems easy to solve it on Recursion, also generate subsets.

\*\* Simple problem and explain it:

-Factorial number:

input: 5

output: 5 \* 4 \* 3 \* 2 \* 1

Answer:

int fact(int num){

if (num == 0)return 1;

return num \* fact(num – 1);

}

explain:

When any function is called from main(), the memory is allocated to it on the stack.

A recursive function calls itself, the memory for a called function is allocated on top of memory allocated to calling function and different copy of local variables is created for each function call.

When the base case is reached, the function returns its value to the function by whom it is called

and memory is de-allocated and the process continues.

If (5 == 0) ? if yes return 1 otherwise then find fact 4

if (4 == 0) ? if yes return 1 otherwise then find fact 3

if (3 == 0) ? if yes return 1 otherwise then find fact 2

if (2 == 0) ? if yes return 1 otherwise then find fact 1

if (1 == 0) ? if yes return 1 otherwise then find fact 0

if (0 == 0) ? if yes return 1 otherwise then find fact -1

\*\* Why should we use the base case ?

because if we didn't write base case, it will be happen overflow.

Problems :

https://codeforces.com/problemset/problem/535/B

https://codeforces.com/group/pbmEl4gUc3/contest/215551/problem/B

https://codeforces.com/group/pbmEl4gUc3/contest/215551/problem/G

https://codeforces.com/group/dfmvO0RN41/contest/227937/problem/C

https://codeforces.com/group/dfmvO0RN41/contest/227937/problem/E

https://codeforces.com/group/dfmvO0RN41/contest/227937/problem/D

https://codeforces.com/group/dfmvO0RN41/contest/227937/problem/B

Resources :

https://www.youtube.com/watch?v=J-BWGN9vUAs&list=PLPSFnlxEu99Ewm9LK4mBLkXeE-p\_Bskha&index=3

**Bitmask**

In computing, numbers are internally represented in binary. This means, where you use an

integer type for a variable, this will actually be represented

internally as a summation of zeros and ones.

Example:

value binary

4 100

9 1001

25 11001

\*\* bitwise :

A B A & B A | B A ^ B

0 0 0 0 0

1 0 0 1 1

0 1 0 1 1

1 1 1 1 0

X ^ 0 = X

X & 1 = X

X | 0 = X

\* NOT in bitmask ( ~ ): flip bits (0 -> 1 , 1 -> 0)

\*\* What literally happens when using this operations?

in computing bits, we will read bits from right to left

and calculate first bit in (A) with first bit in (B)

then second bit in (A) with second bit in (B) and so on ..

Example :

25 & 9 :

16 8 4 2 1

25 1 1 0 0 1

&

9 0 1 0 0 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9 0 1 0 0 1

\*\* shift in Bitmask:

<< shift bits to left

>> shift bits to right

Example:

4 << 1 = 8 (4 \* 2) 1 0 0 0

9 << 2 = 36 (9 \* 4) 1 0 0 1 0 0

4 >> 2 = 1 (4 / 4) 1

9 >> 1 = 4 (9 / 2) 1 0 0

Tricks in Bitmask:

\*\* check if value odd or even:

(value & 1) == (if first bit from right is 1 so is odd otherwise even)

\*\* flip bit from any index:

num ^ (1 << idx - 1);

\*\* set bit (if it is zero change it to one) to any index:

num | (1 << idx - 1);

\*\* get last bit from left:

num & ~(num - 1);

\*\* count bit one in any binary:

while (n){

if ((n & 1) == 1)count\_bit += 1;

n = n >> 1;

}

\*\* Generating Subsets:

the subsets of {1, 2, 3} are {null}, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}

How we find all this ?

int s = (1 << n);

for (int mask = 0; mask < s; i++){

vector<int> v;

for (int j = 0; j < n; j++){

if (1 & (mask >> j) == 1){

v.push\_back(a[j]);

}

}

for (auto x : v)cout << x << " ";

cout << endl;

}

1- Consider The number of possibilities is (2 ^ (size of array))

2- Iterate over all 2^N subsets.

3- Iterate over all bits in the value and find bit one.

if the value is 5 (1 0 1)

if (((5 >> j) & 1) == 1)

1 0 1 j = 0 if (bit == 1) ? push a[0] to vector (j = 0 represent index 0)

0 1 0 j = 1 if (bit == 1) ? push a[1] to vector (j = 1 represent index 1)

0 0 1 j = 2 if (bit == 1) ? push a[2] to vector (j = 2 represent index 2)

5- print subset.

Problems :

https://codeforces.com/problemset/problem/579/A

https://codeforces.com/gym/101810/problem/C

https://codeforces.com/gym/102267/problem/K

https://codeforces.com/problemset/problem/467/B

https://codeforces.com/problemset/problem/550/B

https://codeforces.com/problemset/problem/1097/B

https://codeforces.com/contest/550/problem/B

https://codeforces.com/problemset/problem/1362/C

https://codeforces.com/problemset/problem/1095/C

Resources :

https://www.youtube.com/watch?v=xXKL9YBWgCY

https://codeforces.com/blog/entry/73490