Recursion العودية

هو حل المشكلة بتقسيمها لجزئين

1) جزء نفس المشكلة لحد أصغر 2) جزء معلوم الحل

مشكلة حساب العاملي

1) عاملي حد اصغر 2) الضرب

$$n! = n \times (n-1)!$$

مشكلة حساب الفيبوناتشي

1) فيبوناتشي حد اصغر 2) جمع

$$f(n) = f(n-1) + f(n-2)$$

Recursion العودية

مشكلة حساب العاملي

1) عاملي حد اصغر 2) الضرب

$$n! = n \times (n-1)!$$

+ شرط توقف !!

$$0! = 1$$
$$1! = 1$$

مشكلة حساب الفيبوناتشي

1) فيبوناتشي حد اصغر 2) جمع

$$f(n) = f(n-1) + f(n-2)$$

+ شرط توقف !!

$$f(0) = 1$$
$$f(1) = 1$$

Recursion العودية

مشكلة حساب الفيبوناتشي

```
ll f(ll n)
{
    if(n == 0 || n == 1)
        return 1;
    return f(n-1)+f(n-2);
}
```

$\sim O(2^n)$ *but why?*

مشكلة حساب العاملي

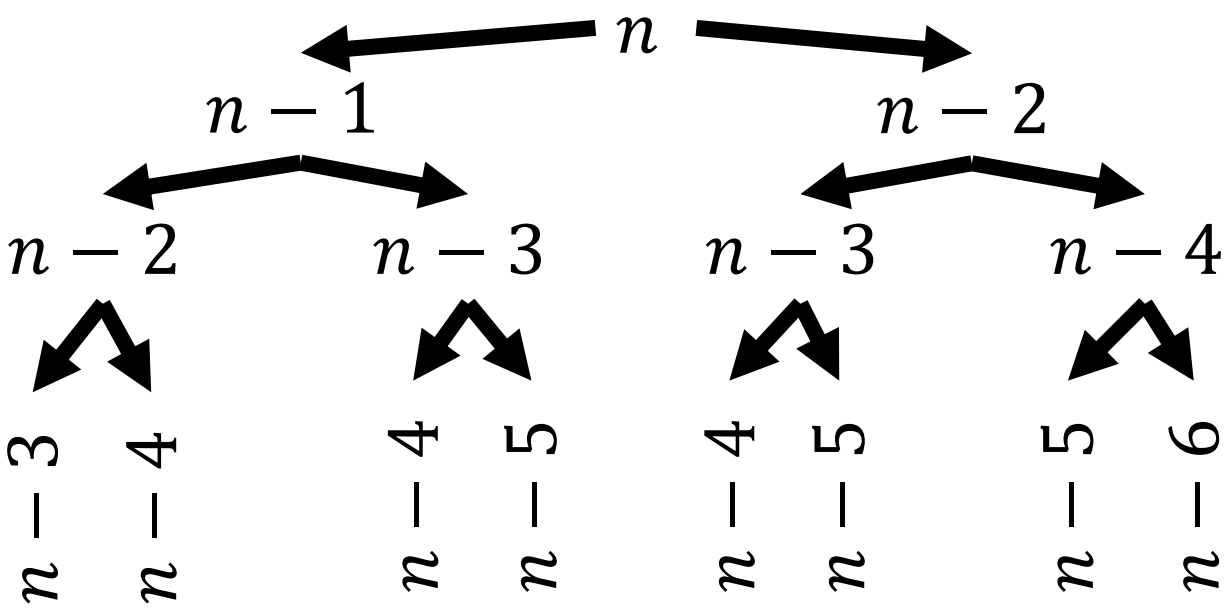
```
ll f(ll n)
{
    if(n == 0 || n == 1)
        return 1;
    return n*f(n-1);
}
```

$O(n)$ *but why?*

Recursion العودية

مشكلة حساب الفيبوناتشي

$\sim O(2^n)$ *but why?*



2^n times

مشكلة حساب العاملي

$O(n)$ *but why?*

$n \rightarrow n-1 \rightarrow n-2 \dashrightarrow 1$

n times

Super Power

$$d^n$$

n even

$$d^{\left\lfloor \frac{n}{2} \right\rfloor} \times d^{\left\lfloor \frac{n}{2} \right\rfloor}$$

n odd

$$d \times d^{\left\lfloor \frac{n}{2} \right\rfloor} \times d^{\left\lfloor \frac{n}{2} \right\rfloor}$$

Ex: $d^4 = d^2 \times d^2$

$d^5 = d \times d^2 \times d^2$

Super Power

$$d^n$$

n even

$$d^{\left\lfloor \frac{n}{2} \right\rfloor} \times d^{\left\lfloor \frac{n}{2} \right\rfloor}$$

n odd

$$d^1 \times d^{\left\lfloor \frac{n}{2} \right\rfloor} \times d^{\left\lfloor \frac{n}{2} \right\rfloor}$$

```
ll power(ll d, ll n){
    if(n==0)
        return 1;
    ll x = power(d, n/2);
    if(n%2==1)
        return x*x*d;
    return x*x;
}
```

$O(\log_2 n)$ *but why?*

Super Power

$O(\log_2(n))$ *but why?*

$n = 10$

$$d^{21} \rightarrow d^{10} \rightarrow d^{5} \rightarrow d^{2} \rightarrow d^{1}$$

$$d^{10} \rightarrow d^{1}$$

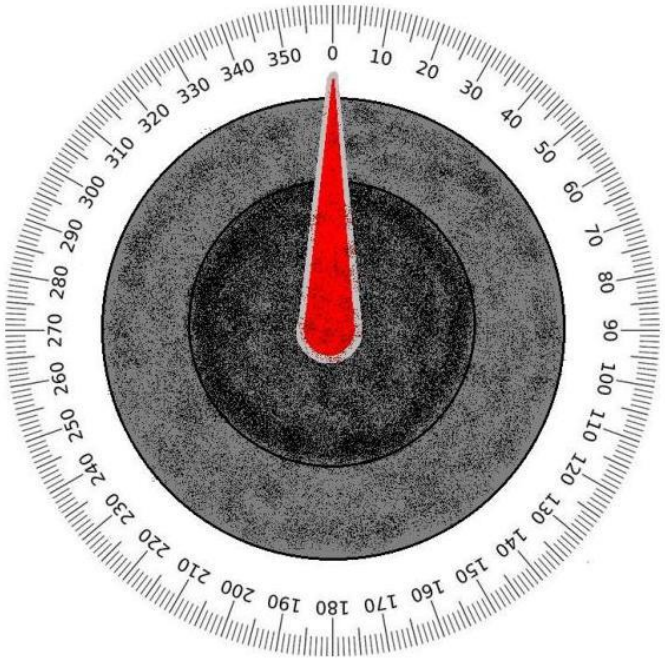
$$d^{2} \rightarrow d^{1}$$

$$d^{1} \rightarrow d^{1}$$

$$n \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \ldots = 1 \Rightarrow \frac{n}{2^m} = 1 \Rightarrow \boxed{m = \log_2(n)}$$

B. Petr and a Combination Lock

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



Max number of rotations is 15

$$\begin{array}{r} 10 \\ 20 \\ 30 \\ \hline 0 \end{array} \quad \begin{array}{r} +\ 10 \\ +\ 20 \\ -\ 30 \\ \hline \end{array}$$

$\Rightarrow$ YES

$$\begin{array}{r} 10 \\ 10 \\ 10 \\ \hline \neq 0 \end{array} \quad \begin{array}{r} 10 \\ 10 \\ 10 \\ \hline \end{array}$$

$\Rightarrow$ NO

B. Petr and a Combination Lock

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

Can sum equal X ??

$$\left\{ \begin{array}{c} \pm a_i \\ \left\{ \begin{array}{c} a_{i+1} \\ a_{i+2} \\ \vdots \\ a_n \end{array} \right. \end{array} \right.$$

Can sum equal

$X + (-a_i)$ OR $X + a_i$??

B. Petr and a Combination Lock

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

```
possbile(0, 0)
```

```
bool possbile(int i, int X) {
    if (i == N)
        return (X % 360) == 0;
    return possbile(i + 1, X + a[i]) || possbile(i + 1, X - a[i]);
}
```

B. Tavas and SaDDas

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

The number n is a **Lucky Number** if their decimal representation doesn't contain digits other than **4**s and **7**s.

Ex: 74, 774, 47, 44, 777….

If we **sort** all lucky numbers in **increasing** order, what's the **1**-based index of **n**?

Where **n** is **Lucky Number**

Ex: 4 → 1

7 → 2

444 → 7

B. Tavas and SaDDas

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

```
set <ll> numbers;
void f(ll n, ll i){
   if(i==10){
      return;
   }else{
      numbers.insert((n*10)+4);
      numbers.insert((n*10)+7);
      f((n*10)+4, i+1);
      f((n*10)+7, i+1);
   }
}
```

```
f(0, 0);
```

B. Lucky Numbers (easy)

https://codeforces.com/problemset/problem/96/B

The number n is a **Super Lucky Number** if it is a **Lucky Number** and contains **equal** amount of digits 4 and 7.

Ex: 74, 47, 4747, 774474….

B. Lucky Numbers (easy)

https://codeforces.com/problemset/problem/96/B

Q: you will get number n.. Find **minimum** number x so that x is a **<u>Super</u> Lucky Number** and $x \geq n$:

Input: 4500
Output: 4747

$1 \leq n \leq 10^9$

Input: 70
Output: 74

B. Lucky Numbers (easy)

https://codeforces.com/problemset/problem/96/B

```
ll ans = 1e18;
void f(ll sum, ll n4, ll n7)
{
    if(sum > 1e10)
        return;
    if(sum >= n && n4 == n7)
        ans = min(sum, ans);
    f(sum*10+4, n4+1, n7);
    f(sum*10+7, n4, n7+1);
}
```

```
f(0, 0, 0);
```

B. Lucky Numbers (easy)

https://codeforces.com/problemset/problem/96/B

```
long long f(long long sum, long long n4, long long n7)
{
    if(sum > 1e10)
        return 1e10;
    if(sum >= n && n4 == n7)
        return sum;
    return min(f(sum*10+4, n4+1, n7), f(sum*10+7, n4, n7+1));
}
```

```
f(0, 0, 0);
```

B. Preparing Olympiad

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

```
#include<bits/stdc++.h>
using namespace std;
int n, L, R, x;
int a[1000];
vector < vector<int> > subsets;
int solve(int index, int sum, int mn, int mx, vector<int>& cur) {
    // base-case
    if (index == n) {
        if (sum >= L && sum <= R && mx - mn >= x) {
            subsets.push_back(cur);
            return 1;
        }
        return 0;
    }
    // calls
    // pick
    cur.push_back(a[index]);
    int pick = solve(index + 1, sum + a[index], min(mn, a[index]), max(mx, a[index]), cur);
    cur.pop_back();
```

B. Preparing Olympiad

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

```
 // leave
    int leave = solve(index + 1, sum, mn, mx, cur);
    return pick + leave;
}

int main(){
    ios::sync_with_stdio(0);
    cin >> n >> L >> R >> x;
    for (int i = 0; i < n; ++i)
        cin >> a[i];
    vector <int> cur;
    cout << solve(0, 0, 1e9, -1e9, cur) << '\n';
    cout << "subsets: \n";
    for (vector<int> vec: subsets) {
        for (int x : vec) cout << x << ' ';
        cout << '\n';
    }
    return 0;
}
```

D. a-Good String

https://codeforces.com/problemset/problem/1385/D

String S is a $\boldsymbol{c}$-Good String if:

- The First half if it is a $(\boldsymbol{c}+\mathbf{1})$-Good String and the second half is "$cccc\ldots$"
- The Second half if it is a $(\boldsymbol{c}+\mathbf{1})$-Good String and the first half is "$cccc\ldots$"
- String S is one string character c

Ex: String $S = "\boldsymbol{eeeehgff}$" is $\boldsymbol{e}$-**Good String**

D. a-Good String

https://codeforces.com/problemset/problem/1385/D

$$S = \text{"}\underbrace{\boldsymbol{e}\ \boldsymbol{e}\ \boldsymbol{e}\ \boldsymbol{e}\ \underbrace{\underbrace{\boldsymbol{h}\ \boldsymbol{g}}_{\text{g-Good}}\ \boldsymbol{f}}_{\text{f-Good}}\ \boldsymbol{f}}_{\text{e-Good String}}\text{"}$$

D. a-Good String

https://codeforces.com/problemset/problem/1385/D

$$S = \text{"}\underbrace{\underbrace{\underbrace{\boldsymbol{b}\ \boldsymbol{b}\ \underbrace{\boldsymbol{c}\ \boldsymbol{d}}_{\text{c-Good}}}_{\text{b-Good}}\ \boldsymbol{a}\ \boldsymbol{a}\ \boldsymbol{a}\ \boldsymbol{a}}_{\text{a-Good String}}\text{"}$$

D. a-Good String

https://codeforces.com/problemset/problem/1385/D

Q: what is the minimum character we need to change to make S a $\boldsymbol{a}$-Good String

Ex: $S = "\boldsymbol{b\ o\ c\ d\ a\ a\ z\ a}"$

$\Downarrow$ 2 Changes

$S = "\boldsymbol{b\ b\ c\ d\ a\ a\ a\ a}"$

Size of the S $\leq 131\,072$

D. a-Good String

https://codeforces.com/problemset/problem/1385/D

```
int solve(int l,int r,char ch){
    if(l==r) return ch!=s[l];
    int mid=(l+r)/2;
    int cntl=0,cntr=0;
    for(int i=l;i<=r;i++){
        if(i<=mid) cntl+=(ch!=s[i]);
        else cntr+=(ch!=s[i]);
    }
    return min(cntl+solve(mid+1,r,ch+1),cntr+solve(l,mid,ch+1));
}
```

```
cout<<solve(0,n-1,'a')<<endl;
```

next_permutation

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    ios::sync_with_stdio(0);
    int n;  cin >> n;
    vector <int> prem;
    for (int i = 1; i <= n; ++i)
        prem.push_back(i);
    do {
        for (int x: prem)
            cout << x << ' ';
        cout << '\n';
    } while (next_permutation(prem.begin(), prem.end()));

    return 0;
}
```