

# 数位dp

---

## 人员

---

左子毅、杨洋、赵清航、于珈浩、刘佳赫、王崇宇、朱奕鸣、刘子淇 到课

## 作业检查

---

左子毅 已完成

杨洋 已完成

赵清航 未完成

于迦浩 未完成

刘佳赫 已完成

王崇宇 已完成

朱奕鸣 上周请假

刘子淇 已完成

## 作业

---

<https://www.luogu.com.cn/contest/175159>

A、B、C 题要求完成

下周要求15分钟内默写 A、B 题，要求同学们在家复习

## 课堂表现

---

今天讲的数位dp有些难度，同学们可能只能理解一部分，课上都有些懵，课下一定要好好复习

## 课堂内容

---

数位dp

求  $1 \sim n$  中有多少符合条件的数

设  $n$  位  $a[n-1], a[n-2], a[n-3], \dots, a[0]$

可以一位一位的看

最高位取  $0 \sim a[n-1]-1$  时, 后面可以任意取

最高位取  $a[n-1]$  后, 第二位可以取  $0 \sim a[n-2]-1$ , 再后面可以任意取

最高位取  $a[n-1]$ , 第二位取  $a[n-2]$  后,  $\dots$  (相同的处理方式)

**注意: 前导0的问题需要特殊考虑**

P2657 [SCOI2009] windy 数

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 15;
int f[maxn][maxn]; // f[i][j]: 第 i 位填 j 时，一共有多少 windy 数

void init() {
    for (int i = 0; i <= 9; ++i) f[1][i] = 1;

    for (int i = 2; i < maxn; i++) {
        for (int j = 0; j <= 9; j++) { // 第 i 位填 j
            for (int j2 = 0; j2 <= 9; j2++) {
                if (abs(j-j2) >= 2) f[i][j] += f[i-1][j2];
            }
        }
    }
}

int dp(int n) { // 1~n 中 windy 数的数量
    vector<int> nums;
    while (n) nums.push_back(n%10), n/=10;

    int res = 0, last = -2;
    for (int i = (int)nums.size()-1; i >= 0; i--) {
        int x = nums[i];
        for (int j = 0; j < x; j++) {
            if (i==(int)nums.size()-1 && !j) continue;
            if (abs(j-last)>=2) res += f[i+1][j];
        }
        if (abs(x-last) < 2) break;
        last = x;
        if (!i) ++res;
    }

    for (int i = (int)nums.size()-2; i >= 0; i--) {
        for (int j = 1; j <= 9; ++j) res += f[i+1][j];
    }
    return res;
}

int main()
{
    init();
    int l, r; cin >> l >> r;
    cout << dp(r) - dp(l-1) << endl;
    return 0;
}

```

P2602 [ZJOI2010] 数字计数

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 15;
LL f[maxn][maxn][maxn]; // f[i][j][k]: 第 i 位填 j 时, k 的数量

LL wPow(int a, int k) {
    LL res = 1;
    while (k --) res *= a;
    return res;
}

void init() {
    for (int i = 0; i <= 9; ++i) f[1][i][i] = 1;

    for (int i = 2; i < maxn; i++) {
        for (int j = 0; j <= 9; j++) {
            for (int k = 0; k <= 9; ++k) {
                for (int j2 = 0; j2 <= 9; ++j2) f[i][j][k] += f[i-1][j2][k];
                if (j == k) f[i][j][k] += wPow(10, i-1);
            }
        }
    }
}

LL calc(vector<int>&nums, int pos) {
    LL res = 0;
    for (int i = pos; i >= 0; --i) res = res*10 + nums[i];
    return res;
}

LL dp(LL n, int k) {
    if (!n && !k) return 1;
    vector<int> nums;
    while (n) nums.push_back(n%10), n /= 10;

    LL res = 0;
    for (int i = (int)nums.size()-1; i >= 0; --i) {
        int x = nums[i];
        for (int j = 0; j < x; ++j) res += f[i+1][j][k];
        if (x == k) {
            LL t = calc(nums, i-1) + 1;
            res += t;
        }
    }

    if (!k) {
        for (int i = (int)nums.size()-1; i >= 1; --i) res -= wPow(10, i);
    }
}

```

```
    }  
    return res;  
}  
  
int main()  
{  
    init();  
    LL l, r; cin >> l >> r;  
    for (int i = 0; i <= 9; ++i) {  
        cout << dp(r, i) - dp(l-1, i) << " ";  
    }  
    cout << endl;  
    return 0;  
}
```