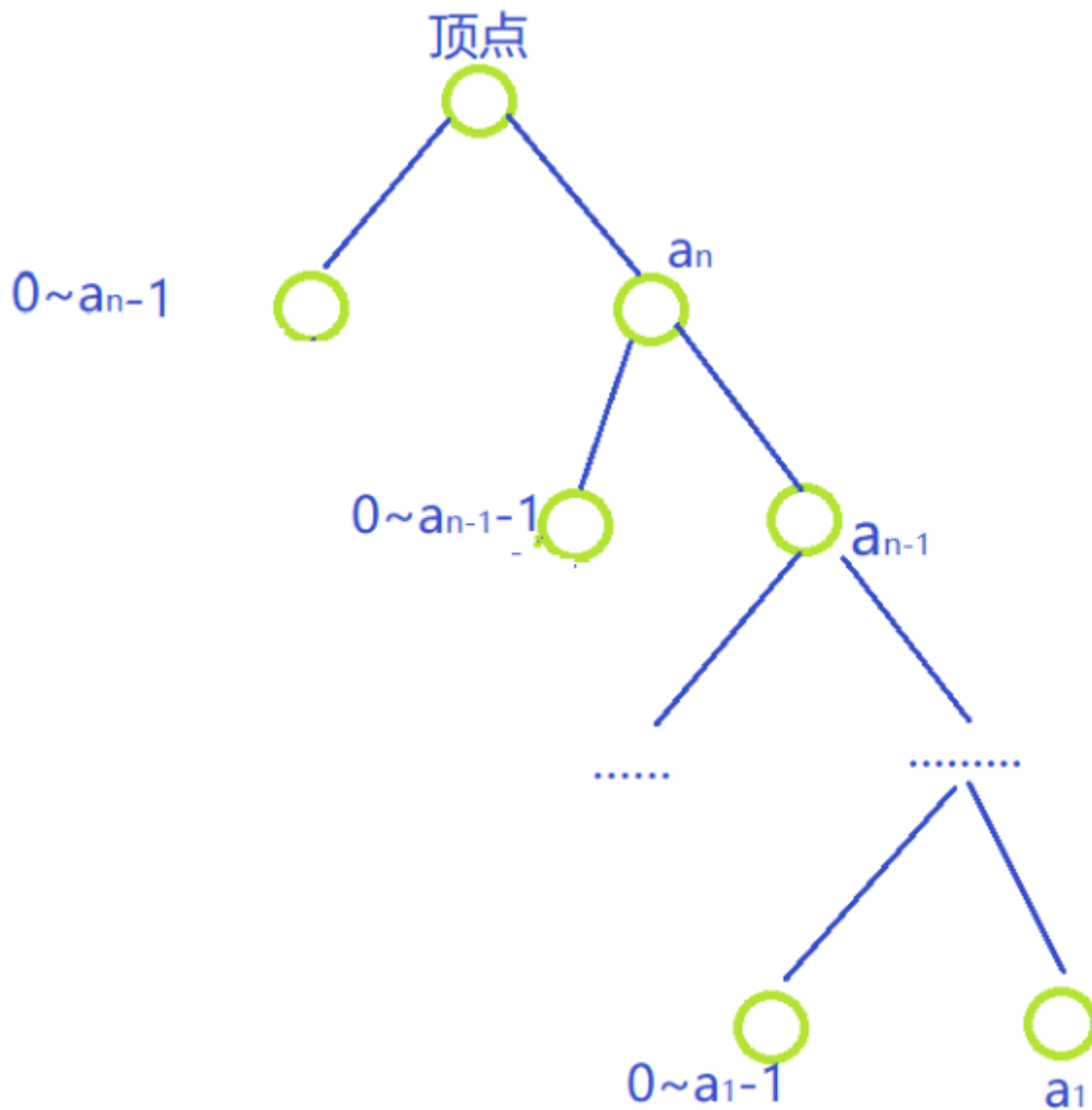


方法：先更具题意进行预处理，一般是排列组合
在更具每一位进行分类讨论



```
int f[N][N]; // f[i][j] 表示是预处理的
void init(){
    for(int i=0; i< N ;i ++){
        for(int j =0; j<= i ;j++){
            if(!j) f[i][j] =1;
            else f[i][j] =f[i-1][j] +f[i-1][j-1];
        }
    }
}
int dp(int n){
    if( ) return 0; //特判
    vector<int> nums; //存放n在B进制下的每一位
    //把n在B进制下的每一位单独拿出来
    while(n) nums.push_back( n% B) , n/= B;
    int res = 0; //答案: [0,n]中共有多少个合法的数
```

```

//last在数位dp中存的是：右边分支往下走的时候保存前面的信息
int last = 0;
//从最高位开始遍历每一位
for(int i = nums.size()-1; i >= 0; i--){
    int x = nums[i]; //取当前位上的数
    if(    ){ //左右分支条件
        res += f[i][K - last]; //加上左分支
        if(    ){
            //右分支判断
            if(K - last - 1 >= 0) res += f[i][K - last - 1];
            //右分支走完
            break;
        }
        //右分支没走完
    } else {
        last++;
        //维护左分支
        if(    ) break; //左分支超出题意
    }

}
//上面处理完了这棵树的**所有**左分支，就剩下最后一位最后一种右分支的情况
if(i == 0 && last == K) res++;
}
return res;
}

```