

镇中110周年校庆赛（小学组）题解 - 博客 - jcsy01 的博客

笔记本： 待整理资料

创建时间： 2021/5/8 20:11

URL: <http://www.zhimaoi.cn/blogof/jcsy01/blog/219>

镇中110周年校庆赛（小学组）题解

2021-05-05 22:16:26 By [jcsy01](#)

水晶球题解

用三分法（三分法：每次平均分成三堆，称一次，可以找出硬币在三堆中的一堆。）

先不考虑 m ，则只要称 $\log_3 n$ （取下整）次。

然后考虑 m ，把 n 分成 $m, m, n-2m$ 三份，每次筛掉 $2m$ 个。（不用考虑 m 的情况也有，当 $n-2m \leq m+1$ 时）

如何筛选这 $2m$ ，可以用除法进行优化，即 $(n-3)*(m-1)/2m$ （取下整）

记得开 unsigned long long

```
cin>>n>>m;
t=(n-3*m-1)/(2*m)+1;
if(3*m+1<n){
    n-=t*2*m;
    ans=t;
}
int i=1;
while(n>i){
    i*=3;
    ans++;
}
cout<<ans<<endl;
```

Dove与Sweet 题解

60pts思路

注意输入时要用字符串！

直接暴力枚举 $2 \sim n$ ，求数位差即可。

```

#include<bits/stdc++.h>
using namespace std;
long long int n,ans,mod=1914270647;
long long int a(long long int x){
    long long int k=0;
    while(x){
        k+=x%10%mod;
        x/=10;
    }
    return k;
}
int main()
{
    cin>>n;
    for(int i=2;i<=n;i++){
        ans+=abs(a(i)-a(i-1))%mod;
    }
    cout<<ans;
}

```

100pts思路

将每一次进位的结果（如1 ~ 10, 91 ~ 100 记录，可以发现如下规律：

$$k_1 = 1, k_2 = 17, \dots, k_n = k_{n-1} * 10 + 9$$

$$\text{所以, } \text{Ans} = \sum_{i=0}^{n.\text{size}()-1} (n_i - '0') * k_{n.\text{size}()-i}$$

```

#include<bits/stdc++.h>
using namespace std;
long long ans,mod=1914270647,k[30000010];
string n;
int main()
{
    ios_base::sync_with_stdio(false);
    cin>>n;
    k[1]=1;k[2]=17;
    for(int i=3;i<=n.size()+1;i++){
        k[i]=(k[i-1]*10+9)%mod;
    }
    for(int i=n.size()-1,j=1;i>=0;i--,j++){
        ans=(ans+(n[i]-'0')*k[j])%mod;
    }
    cout<<ans-1;
}

```

更快的标程，核心式子是一样的

```

#include<bits/stdc++.h>
using namespace std;
const long long mod=1914270647;
string s;
int n;

```

```

long long ans,t;

int main()
{
    ios_base::sync_with_stdio(false);
    cin>>s;
    n=s.length();
    for(int i=0;i<(n-2);i++)
    {
        t=(t*10+s[i]-'0')%mod;
        ans=(ans+t*9)%mod;
    }
    t=(t*10+s[n-2]-'0')%mod;
    ans=(ans+t*8)%mod;
    ans=(ans-t+mod)%mod;
    t=(t*10+s[n-1]-'0')%mod;
    ans=(ans+t-1+mod)%mod;
    cout<<ans<<endl;
    return 0;
}

```

A国和B国的战争

这题代码是真的短啊。

首先要把题目转化一下。（最重要的一步）

因为最大只能跳 m 个单位距离，所以一个士兵肯定会踩到任意连续的 m 个位置中的一个浮岛。

对于所有连续 m 个位置的区间，它最多通过士兵数量 = 区间内浮岛的和，也就是 $S_{l+m-1} - S_{l-1}$ （ S 为浮岛数量的前缀和， l 为区间左端点）

所以答案就是对所有长度为 m 的区间中最多通过士兵数量取 \min 。

（注意开 long long。）

代码：

```

#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1000005;
int n,m;
ll a[N],s[N],ans=1e15;
int main(){
    scanf("%d%d",&n,&m);
    for(int i=1;i<n;i++){
        scanf("%lld",&a[i]);
        s[i]=s[i-1]+a[i];
    }
    for(int i=1;i<=n-m;i++)
        ans=min(ans,s[i+m-1]-s[i-1]);
    printf("%lld",ans);
}

```

```
return 0;  
}
```

鸽子的作文 题解

dp题

30pts

$dp[i][j][k]$ 代表对于前 i 个字符，修改 j 次，最后一个字符是 k ，最少可以划分成几个连续段。

若 k =输入字符串 $s[i]$ ，不用修改，修改次数不变：

$dp[i][j][k] = \min(dp[i-1][j][k], // \text{前一个字符也是} k, \text{连续段数量不变} \min(dp[i-1][j][\text{非} k]) + 1) // \text{前一个字符不是} k, \text{连续段数量加} 1$ 否则，修改次数会增加1：
 $dp[i][j][k] = \min(dp[i-1][j-1][k], // \text{前一个字符也是} k, \text{连续段数量不变} \min(dp[i-1][j-1][\text{非} k]) + 1) // \text{前一个字符不是} k, \text{连续段数量加} 1$ 最终答案是 $\min(dp[n][j][k])$

100pts

滚动数组把dp的第一维压掉（为了方便理解，在题解里把 i 留着）

我们发现求 $\min(dp[i][j][\text{非} k])$ 很花时间

用 $\text{mindp}[i][j][0]$ 代表 $dp[i][j][k]$ 中最小的

用 $\text{mindp}[i][j][1]$ 代表 $dp[i][j][k]$ 中次小的

如果 $dp[i][j][k]$ 中最小值出现次数不止一次，那么 $\text{mindp}[i][j][1] = \text{mindp}[i][j][0]$

求 mindp 不需要sort（这个不用讲了吧）

如果 $dp[i][j][k] = \text{mindp}[i][j][0]$ ，那么 $\min(dp[i][j][\text{非} k]) = \text{mindp}[i][j][1]$

否则 $\min(dp[i][j][\text{非} k]) = \text{mindp}[i][j][0]$

于是计算单个 $dp[i][j][k]$ 的时间缩小了26倍

最终答案是 $\min(\text{mindp}[n][j][0])$

阶乘鸽 题解

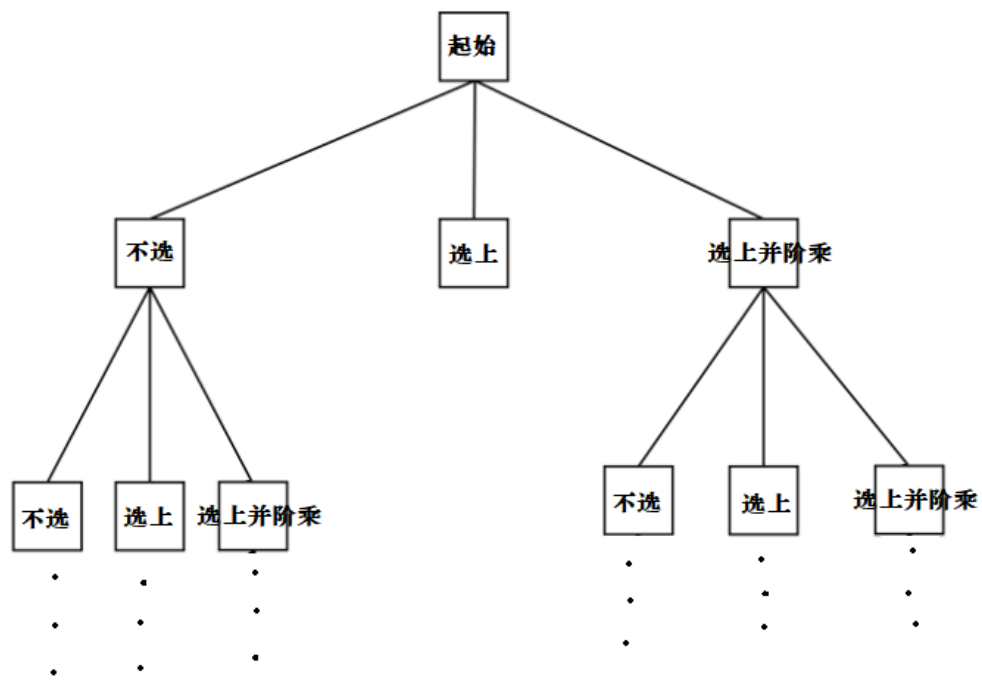
看题面，发现对于每一个数都有 **不选**，**选上**，**选上并阶乘** 这三种状态。

如果直接 dfs 暴力 $\Theta(3^n)$ ，获得 30 pts

需要进一步优化，我们可以考虑 Meet In The Middle 复杂度优化为 $\Theta(2 \times 3^{\frac{n}{2}})$ （假定 Map $\Theta(1)$ ）。

优化的原理：

如果是 dfs：整体会呈现一个如三角形的递归



如果是 Meet In The Middle 整体呈现为一个菱形（画图渣渣

针对匹配，不难发现，能够匹配的是

$$k - \text{used} \\ \sum_{i=0} m[i][S - \text{sum}]$$

那么，就做完啦。

```
#include<bits/stdc++.h>
#define int long long

using namespace std;
const int Maxn = 30+50;
const int MaxA = 20;

unordered_map<int,int> m[Maxn];

int n,k,S;
int fact[Maxn],a[Maxn];
int Ans_=0;

inline void dfs(int flag,int now,int End,int sum,int used)
{
    if(used > k || sum > S) return;
    if(now > End)
    {
        if(flag==1) ++m[used][sum];
        else
        {
            for(register int i=0;i<=k-used;++i)
                Ans_ += m[i][S-sum];
        }
        return ;
    }

    dfs(flag,now+1,End,sum,used);
    dfs(flag,now+1,End,sum+a[now],used);
    if(a[now] <= MaxA) dfs(flag,now+1,End,sum+fact[a[now]],used+1);
}

signed main()
{
    fact[1]=1;
    for(register int i=2;i<=20;++i) fact[i]=fact[i-1]*i;

    scanf("%lld%lld%lld",&n,&k,&S);
    for(register int i=1;i<=n;++i) scanf("%lld",&a[i]);

    register int mid=n+1>>1;
    dfs(1,1,mid,0,0),dfs(2,mid+1,n,0,0);

    return !printf("%lld\n",Ans_);
}
```

评论 ☐

caiyuhao

☐好评 ☐差评 [0]

T5图挂了

2021-05-06 07:13:45 [回复](#)

评论回复

jcsy01: 已修复

2021-05-06 15:38:48 [回复](#)

hutianshuo: 回复 @jcsy01: 下面的图反了

2021-05-06 18:52:20 [回复](#)

gonghengjie: 回复 @jcsy01: 下面一半字怎么回事

2021-05-06 21:58:43 [回复](#)

Dove: 那个图本来就是反的, 因为lazy [doge]

2021-05-07 11:53:31 [回复](#)

jcsy01: 回复 @gonghengjie: 能理解就行

2021-05-07 19:00:01 [回复](#)

yuanshiji

☐好评 ☐差评 [0]

@you 最后一题错了, 不能运行

2021-05-06 21:10:55 [回复](#)

评论回复

jcsy01: 看看是不是用的c++11

2021-05-07 08:50:50 [回复](#)

yuziteng1

☐好评 ☐差评 [0]

第一题得了30分

回复

评论回复

YZH: hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh

回复

发表评论

可以用 @mike 来提到 mike 这个用户，mike 会被高亮显示。如果你真的想打“@”这个字符，请用“@@”。

想使用表情？参见[表情指南](#)。

内容