

Codeforces Raif Round 1 (Div. 1 + Div. 2)

[比赛链接](#)

E. Carrots for Rabbits

题意

给定 n 个数 a_i 要求数列 b 满足

1. $\sum_{j=1}^{k_i} b_{i,j} = a_i$
2. $\sum_{i=1}^n k_i = k$
3. $k_i, b_{i,j} \in \mathbb{N}^+$

要求最小化 $\sum b_{i,j}^2$

题解

考虑函数 $f(i,j)$ 表示将 a_i 分成 j 份时的答案，显然均分为最佳策略。

考虑维护 $f(1,k_1), f(2,k_2) \dots f(n,k_n)$ 初始时 $k_i=1$

每次操作选择一个 i 使得 $f(i,k_i) \rightarrow f(i,k_i+1)$ 显然选择减小值最大的数最佳，可以优先队列维护。

共进行 k 次操作(包括初始 n 次入队操作)，时间复杂度 $O(k \log n)$

```
LL f(int a,int b){
    return 1LL*(a/b)*(a/b)*(b-a%b)+1LL*(a/b+1)*(a/b+1)*(a%b);
}
struct node{
    int a,k;
    LL d;
    node(int a=0,int k=0){
        this->a=a;
        this->k=k;
        d=f(a,k)-f(a,k+1);
    }
    bool operator < (const node &b) const{
        return d<b.d;
    }
};
priority_queue<node> q;
int main()
{
    int n=read_int(),k=read_int();
    LL ans=0;
```

```
_for(i,0,n){
    int a=read_int();
    q.push(node(a,1));
    ans+=1LL*a*a;
}
while(n<k){
    node temp=q.top();q.pop();
    ans-=temp.d;
    q.push(node(temp.a,temp.k+1));
    n++;
}
enter(ans);
return 0;
}
```

G2. Lucky Numbers (Hard Version)

题意

给定一个表，代表每个位的权值。定义一个数的权值和为每个位的权值和。

接下来若干询问，要求将给定的数 n 拆分成 k 个数，使得拆分得到的数的权值和最大。

数据范围保证 $n, k \leq 10^6$



题解

先考虑将 n 全部拆分为每个位均为 $0, 3, 6, 9$ 的数的情况。发现可以先计算出每个位的情况，最后将不同位任意组合，即可得到 k 个数。

而对与每个位，发现可以将 6 视为 2 个 3 ， 9 视为 3 个 3 。于是等价于每个位均可以选择至多 $3k$ 个价值为 F_i 重量为 3×10^i 的物品。

最后只需要所有位的物品重量和等于 n 即可。接下来考虑不能使得最后物品重量和等于 n 的情况。

对每个位的情况，可以证明拆分中至多有一个数的该位不为 3 的倍数。否则如果有两个数该位不为 3 的倍数，记为 a, b

若 $a+b \geq 9$ 则将这两位修改为 $a+b-9, 9$ 否则，将这两位修改为 $0, a+b$ 总能使得最终权值和不减。

于是不妨将所有不为 3 的倍数集中到一个数上面，等价于将 n 拆分为一个任意数和 $k-1$ 个每个位均为 3 的倍数的数。

于是先预处理出 $0 \sim 99999$ 的权值作为背包的基础值，然后跑二进制优化的 01 背包即可。

总时间复杂度 $O(18v \log k + q)$ 其中 $v=10^6$

```
const int MAXN=1e6,MAXC=205;
LL dp[MAXN],v[MAXC],w[MAXC];
int f[10];
int main()
{
    int k=read_int(),cnt=0;
    _for(i,0,6)f[i]=read_int();
    _for(i,1,MAXN){
        for(int j=0,t=i;j<6;j++,t/=10)
            dp[i]+=1LL*(t%10%3==0)*(t%10/3)*f[j];
    }
    k=3*(k-1);
    for(int i=0,pow_10=1;i<6;i++,pow_10*=10){
        int t=k;
        for(int j=1;j<=t;j<=1){
            v[cnt]=1LL*f[i]*j,w[cnt++]=3LL*pow_10*j;
            t-=j;
        }
        if(t)v[cnt]=1LL*f[i]*t,w[cnt++]=3LL*pow_10*t;
    }
    _for(i,0,cnt)
        for(int j=MAXN-1;j>=w[i];j--)
            dp[j]=max(dp[j],dp[j-w[i]]+v[i]);
    int q=read_int();
    while(q--){
        enter(dp[read_int()]);
    }
    return 0;
}
```

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:cf_raif_div_1_div_2

Last update: 2020/10/20 16:21