

## Solution DAY 11/10

### Solution bài 1 : Hệ thống tiền tệ

$F[i]$  đại diện cho số lượng tối thiểu với lớn nhất là  $i$

$F[i]$  rõ ràng là được chuyển từ thừa số  $j$  của  $i$ ,  $F[j]$

Phương trình chuyển tiếp:  $F[i] = \min (F[j] - \sum [a[k] / i] * (i / j - 1))$

Với  $a[k] / i$  có nghĩa là cần vài đồng xu có giá trị  $i$ , những đồng xu này có giá trị  $(i / j)$  đồng xu có giá trị  $j$ , vì vậy nó có thể ít hơn  $(i / j - 1)$

Một tối ưu hóa là khi liệt kê  $j$ , có thể bỏ qua một số, chỉ cần liệt kê  $i / p$  ( $p$  là thừa số nguyên tố của  $i$ )

### Solution bài 2 : Xây dựng tuyến đường

Gọi  $dp[i]$  là hiện tại chi phí xây dựng nhà ga  $i$ , và sau đó chúng tôi tìm công thức đệ quy:

$$dp[i] = \min(dp[j] + m + \sum_{j < k \leq i} (R[k] * (T[k] - T[i])))$$

Sau đó, xem xét khai triển, chúng ta đặt  $sumt[]$  là tiền tố,  $sum[]$  là tổng tiền tố của  $R[]$ , rồi viết công thức:

$$dp[i] = \min(dp[j] + m + sumt[i] - sumt[j] - T[i] * sum[i] + T[i] * sum[j])$$

Sau đó, loại bỏ min không liên quan và viết công thức:

$$dp[i] = \min(dp[j] - sumt[j] + T[i] * sum[j]) + m - sumt[i] - T[i] * sum[i]$$

### Solution bài 3 : Vận chuyển hàng hóa

Đầu tiên, sử dụng SPFA để xử lý đường đi ngắn nhất từ điểm bắt đầu đến điểm cuối trong ngày thứ  $i$  đến ngày thứ  $j$  và sau đó có thể giải được bằng  $Dp$ .

$dp[i]$  đại diện cho chi phí tối thiểu cho đến ngày  $i$

$$dp[i] = \min(dp[j] + f[j+1][i] * (i - j) + k) \quad (0 \leq j < i)$$

$f[i][j]$  Đại diện cho đường đi ngắn nhất từ ngày thứ  $i$  đến ngày thứ  $j$ , có thể được xử lý trước.

## CODE THAM KHẢO

### Bài 1.

```
#include <bits/stdc++.h>
#define T 100001
using namespace std;

int f[T],a[55],v[T],prime[T];
int n,tot,top,ans,mx=0;
void __prime()
{
    for(int i=2;i<T;i++)
    {
        if(!v[i]) prime[++top]=i;
        for(int j=1;j<=top&&prime[j]*i<T;j++)
        {
            v[i*prime[j]]=1;
            if(i%prime[j]==0)break;
        }
    }
}
int main()
{
    freopen("MONEY.inp","r",stdin) ;
    freopen("MONEY.out","w",stdout) ;
    ios_base :: sync_with_stdio(0) ;
    __prime();
    cin>>n;
    for(int i=1;i<=n;i++)
        { cin>>a[i],tot+=a[i],mx=max(mx,a[i]); }
    ans=f[1]=tot;
    for(int i=2;i<=mx;i++)
    {
        f[i]=tot;
        int p=i;
        for(int j=1;j<=top;j++)
            if(p%prime[j]==0)
            {
                int x=i/prime[j];
                int sum=0;
                for(int k=1;k<=n;k++) sum+=(a[k]/i)*(prime[j]-1);
                f[i]=min(f[i],f[x]-sum);
                while(p%prime[j]==0)p/=prime[j];
                if(p==1)break;
            }
        ans=min(ans,f[i]);
    }
    cout<<ans;
```

```

    return 0;
}

```

Bài 2:

```

#include <bits/stdc++.h>
#define f first
#define s second
using namespace std ;

int const N = 4e4+5;
int const int_lim = 1e9+7 ;
long long const mod = 1e9+7ll ;
long long const ll_lim = 1e18 ;
int n ;
long long m ;
pair < long long , long long > a[N] ;
long long s[N] , f[N] , F[N] , sm[N] ;
int l=1 ;
bool maximize(long long &x, long long y){if (x<y){x=y;return 1;}return 0;}
bool minimize(long long &x, long long y){if (x>=y){x=y;return 1;}return 0;}

```

```

void solve ()
{
    sort ( a+1 , a+1+n ) ;
    F[0] = 0ll ;
    f[0] = 0ll ;
    sm[0] = 0ll ;
    s[0] = 0ll ;
    for ( int i=1 ; i<=n ; i++ )
    {
        sm[i] = sm[i-1] + a[i].f * a[i].s ;
        s[i] = s[i-1] + a[i].s ;
    }

    for ( int i=1 ; i<=n ; i++ )
    {
        F[i] = f[i-1] + m ;
        f[i] = f[i-1] + a[i].f * a[i].s ;
        for ( int j=1 ; j<=i ; j++ )
        {
            long long cmp = F[j] + (sm[i] - sm[j-1]) - ( s[i]-s[j-1] ) * a[j].f ;
            if ( minimize ( f[i] ,cmp ) ) l = j ;
        }
    }
    cout << f[n] ;
}

```

```

void inp()
{

```

```

    cin >> n >> m ;
    for ( int i=1 ; i<=n ; i++ )
        cin >> a[i].f >> a[i].s ;
}

int main()
{
    freopen("BUILD.inp","r",stdin) ;
    freopen("BUILD.out","w",stdout) ;
    ios_base :: sync_with_stdio(0) ;
    cin.tie(0) ;
    cout.tie(0) ;

    inp () ;
    solve () ;
    return 0 ;
}

```

Bài 3:

```

#include <bits/stdc++.h>
using namespace std;
const int maxN=500,inf=0x3f3f3f3f;

struct Node
{
    int to,value,next;
}edge[maxN*2+1];
int n,m,K,e,d,x,y,t,p,a,b,dis[maxN+1],f[maxN+1];
int tot,head[maxN+1];
bool flag[maxN+1][maxN+1],vis[maxN+1];
bool mark[maxN+1];

inline void add(int x,int y,int t)
{
    edge[++tot]=(Node){y,t,head[x]}; head[x]=tot;
    edge[++tot]=(Node){x,t,head[y]}; head[y]=tot;
}
inline void spfa()
{
    memset(dis,0x3f,sizeof(dis));
    queue<int> q; q.push(1); dis[1]=0;
    while(!q.empty())
    {
        int x=q.front(); q.pop();
        vis[x]=false;
        for(int i=head[x];i;i=edge[i].next)
            if(!mark[edge[i].to]&&dis[x]+edge[i].value<dis[edge[i].to])
            {
                dis[edge[i].to]=dis[x]+edge[i].value;
            }
    }
}

```

```

        if(!vis[edge[i].to])
        {
            vis[edge[i].to]=true;
            q.push(edge[i].to);
        }
    }
}
}
inline void work()
{
    f[0]=-K;
    for(int i=1;i<=n;i++)
    {
        memset(mark,false,sizeof(mark));
        for(int j=i;j>=1;j--)
        {
            for(int k=1;k<=m;k++)
                if(flag[k][j]) mark[k]=true;
            spfa();
            if(dis[m]>=inf) continue;
            f[i]=min(f[i],f[j-1]+(i-j+1)*dis[m]+K);
        }
    }
    cout<<f[n];
}
int main()
{
    freopen("TRANSPORT.inp","r",stdin) ;
    freopen("TRANSPORT.out","w",stdout) ;
    ios_base :: sync_with_stdio(0) ;
    cin>>n>>m>>K>>e;
    for(int i=1;i<=e;i++)
    {
        cin>>x>>y>>t;
        add(x,y,t);
    }
    cin>>d;
    for(int i=1;i<=d;i++)
    {
        cin>>p>>a>>b;
        for(int j=a;j<=b;j++) flag[p][j]=true;
    }
    memset(f,0x3f,sizeof(f));
    work();
    return 0;
}

```