

# 分数规划

实际上是求一个分式的极值，通常用二分方法来解决。

一般分数问题会有奇怪的限制，比如“分母至少为W”。

## 分数规划

[Dinkelbach算法](#)

[板子](#)

[Dropping tests](#)

[带分母限制的分数规划](#)

[P4377 \[USACO18OPEN\]Talent Show G](#)

[Desert King](#)

[HNOI2009\]最小圈](#)

[习题](#)

[参考资料](#)

## Dinkelbach算法

Dinkelbach算法是分数规划的基本思想，大概为每次用上一轮的答案当作新的L（二分的左端点）来输入，不断的迭代直至收敛。

## 板子

### Dropping tests

```
#include<bits/stdc++.h>
#define inf 0x3f3f3f3f
// #define int long long
using namespace std;
const int N=1007;
const int mod=1e9+7;

int n,k;
double a[N],b[N],c[N];

bool check(double ans){
    for(int i=1;i<=n;i++){
        c[i]=a[i]-b[i]*ans;
    }
    sort(c+1,c+1+n);
    double res=0;
    for(int i=n;i>k;i--){
        res+=c[i];
    }
    if(res>=0) return true;
    return false;
}

signed main(){
```

```

while(cin>>n>>k){
    if(n==0&&k==0) break;
    for(int i=1;i<=n;i++) cin>>a[i];
    for(int i=1;i<=n;i++) cin>>b[i];
    double l=0,r=inf,mid;
    while(r-l>1e-6){
        mid=(l+r)/2;
        if(check(mid)) l=mid;
        else r=mid;
    }
    cout<<int(l*100+0.5)<<"\n";
}
return 0;
}

```

## 带分母限制的分数规划

### P4377 [USACO18OPEN]Talent Show G

使用01背包，dp[n][W]为最大值。

```

#include<bits/stdc++.h>
#define inf 0x3f3f3f3f
using namespace std;
const int N=257;
const int mod=1e9+7;

int n,w,w[N],t[N];
double c[N],dp[1005];

bool check(double x){
    for(int i=1;i<=n;i++) c[i]=(t[i]-w[i]*x);
    for(int i=1;i<=w;i++) dp[i]=-inf;
    for(int i=1;i<=n;i++){
        for(int j=w;j>=0;j--){
            int k=min(w,j+w[i]);
            dp[k]=max(dp[k],dp[j]+c[i]);
        }
    }
    return dp[w]>0;
}

signed main(){
    cin>>n>>w;
    for(int i=1;i<=n;i++){
        cin>>w[i]>>t[i];
    }
    double l=0,r=inf,mid;
    while(r-l>1e-6){
        mid=(l+r)/2;
        if(check(mid)) l=mid;
        else r=mid;
    }
    cout<<(int)(l*1000)<<"\n";
    return 0;
}

```

```
}
```

## Desert King

给定三维的点，两点距离为水平距离，两点连通花费为垂直距离，求一棵生成树使得

$$\frac{\sum cost_i}{\sum len_i} = ans \text{最小}。$$

```
#include<iostream>
#include<cstring>
#include<cmath>
#define inf 0x3f3f3f3f
using namespace std;
const int N=1e3+7;
const int mod=1e9+7;

int n,x[N],y[N],z[N],vis[N];
double dis[N],C[N][N],D[N][N];

double prim(int u,double k){ //MST+分数划分
    memset(vis,0,sizeof(vis));
    int p=1;vis[u]=1;dis[1]=0;
    for(int i=2;i<=n;i++) dis[i]=C[u][i]-D[u][i]*k;
    double Min,ans=0;
    for(int i=2;i<=n;i++){
        Min=inf;
        for(int j=1;j<=n;j++){
            if(!vis[j]&&dis[j]<Min){
                Min=dis[j];
                p=j;
            }
        }
        vis[p]=1;
        ans+=dis[p];
        for(int j=1;j<=n;j++){
            double v=C[p][j]-D[p][j]*k;
            if(!vis[j]&&v<dis[j]) dis[j]=v;
        }
    }
    return ans;
}

signed main(){
    while(cin>>n&&n){
        memset(D,0,sizeof(D));
        memset(C,0,sizeof(C));
        for(int i=1;i<=n;i++){
            cin>>x[i]>>y[i]>>z[i];
        }
        for(int i=1;i<=n;i++){
            for(int j=i+1;j<=n;j++){
                D[i][j]=D[j][i]=sqrt(1.0*(x[i]-x[j])*(x[i]-x[j])+1.0*(y[i]-y[j])*(y[i]-y[j]));
                C[i][j]=C[j][i]=abs(z[i]-z[j]);
            }
        }
    }
}
```

```

    }
    double l=0,r=100,mid;
    while(r-l>=1e-7){
        mid=(l+r)/2;
        if(prim(1,mid)>=0) l=mid;
        else r=mid;
    }
    printf("%.3lf\n",r);
}
return 0;
}

```

### [HNOI2009\]最小圈](#)

在图上求一个环 $C$ 使得  $\frac{\sum_{e \in C} w}{|C|}$  (环上边权之和除以边数) 最小。

二分跑SPFA即可，每次只要找到一个环满足  $\sum (w_i - mid) > 0$  即为合法。

```

#include<bits/stdc++.h>
#define inf 0x3f3f3f3f
using namespace std;
const int N=3007;
const int mod=1e9+7;

struct E{
    int v,nxt;
    double w;
}e[N<<2];

int n,m,u,v,vis[N];
double dis[N],w;
int head[N],cnt=0;
void addedge(int u,int v,double w){
    e[++cnt].v=v;
    e[cnt].w=w;
    e[cnt].nxt=head[u];
    head[u]=cnt;
}

inline bool SPFA(int u,double mid){ //判负环
    vis[u]=1;
    for(int i=head[u];i;i=e[i].nxt){
        int v=e[i].v;
        double w=e[i].w-mid;
        if(dis[u]+w<dis[v]){
            dis[v]=dis[u]+w;
            if(vis[v]||SPFA(v,mid)) return 1;
        }
    }
    vis[u]=0;
    return 0;
}

inline bool check(double mid){
    for(int i=1;i<=n;i++) dis[i]=0,vis[i]=0;
}

```

```

        for(int i=1;i<=n;i++){
            if(SPFA(i,mid)) return 1;
        }
        return 0;
    }

    signed main(){
        cin>>n>>m;
        for(int i=1;i<=m;i++){
            cin>>u>>v>>w;
            addedge(u,v,w);
        }
        double l=-inf,r=inf,mid;
        while(r-l>1e-10){
            double mid=(l+r)/2;
            if(check(mid)) r=mid;
            else l=mid;
        }
        printf("%.81f\n",r);
        return 0;
    }
}

```

## 习题

- [JSOI2016 最佳团体](#)
- [SDOI2017 新生舞会](#)
- [UVa1389 Hard Life](#)

## 参考资料

OI-wiki