# Fenshu

2728

#include<cstdio>

using namespace std;

const int M = 1234;

const int N = 123456789;

int vis[M];

int pre[M];

double dist[M];

double a[M][M],b[M][M];

int n;

double prim(double K){

double totalcost=0, totaldist=0;

double sum=0;

for(int i=0;i<=n;i++){

dist[i]=N\*1.0;

vis[i]=0;

}

dist[1]=0.0;

for(int i=1;i<=n;i++){

int k=0;

for(int j=1;j<=n;j++){

if(vis[j]) continue;

if(dist[j]<dist[k]) k=j;

}

vis[k]=1;

sum+=dist[k];

if(k!=1){//牛顿迭代法，二分法不用

totalcost+=a[pre[k]][k];

totaldist+=b[pre[k]][k];

}

for(int j=1;j<=n;j++){

if(vis[j]) continue;

if(dist[j]>a[k][j]-b[k][j]\*K){

dist[j]=a[k][j]-b[k][j]\*K;

pre[j]=k;

}

}

}

//return sum;//二分法

return totalcost/totaldist; // 牛顿迭代法 ，不断逼近

}

double binary(double l,double r){

while(r-l>1e-6){

double mid=(l+r)/2.0;

if(prim(mid)<0.0)r=mid;

else l=mid;

}

return r;

}

double newton(){

double A=0.0,B=100.0;

while(1){

double B = prim(A);

if(fabs(A-B)<1e-6) break;

A=B;

}

return A;

}

double x[M],y[M],z[M];

double p2(double A,double B){

return (A-B)\*(A-B);

}

int main(){

freopen("1.txt","r",stdin);

//freopen("A.out","w",stdout);

while(cin>>n){

if(n==0) break;

for(int i=1;i<=n;i++){

cin>>x[i]>>y[i]>>z[i];

}

for(int i=1;i<=n;i++){

for(int j=1;j<=n;j++){

b[i][j]=sqrt(p2(x[i],x[j])+p2(y[i],y[j]));

a[i][j]=fabs(z[i]-z[j]);

}

}

printf("%.3lf\n",newton());

}

return 0;

}