**算法训练 旅行家的预算**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　一个旅行家想驾驶汽车以最少的费用从一个城市到另一个城市（假设出发时油箱是空的）。给定两个城市之间的距离D1、汽车油箱的容量C（以升为单位）、每升汽油能行驶的距离D2、出发点每升汽油价格P和沿途油站数N（N可以为零），油站i离出发点的距离Di、每升汽油价格Pi（i=1，2，……N）。计算结果四舍五入至小数点后两位。如果无法到达目的地，则输出“No Solution”。

输入格式

　　第一行为4个实数D1、C、D2、P与一个非负整数N；  
　　接下来N行，每行两个实数Di、Pi。

输出格式

　　如果可以到达目的地，输出一个实数（四舍五入至小数点后两位），表示最小费用；否则输出“No Solution”（不含引号）。

样例输入

275.6 11.9 27.4 2.8 2  
102.0 2.9  
220.0 2.2

样例输出

26.95

锦囊1

贪心。

锦囊2

使用假装加满油的方法，记录下自己油箱里面有哪些油以及价格是多少。 到了新的加油站再用新的油把油箱加满。如果比较便宜，就把自己油箱里面的贵的油以原价“退回”到原来地方。 最终到达目的地把所剩的油“退回”。

本题的C++参考代码如下：

#include<cstdio>

using namespace std;

int main(){

int n;

double d1,c,d2,p0;

double p[1000],d[1000];

scanf("%lf%lf%lf%lf%d",&d1,&c,&d2,&p0,&n);

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

scanf("%lf%lf",&d[i],&p[i]);

}

d[0]=0;

p[0]=p0;

d[n+1]=d1;

p[n+1]=0;

double sum=0,ct=0;

bool flag=1;

for(int i=0,j;i<=n;i=j){

flag=0;

if(d[i+1]-d[i]>c\*d2)break;

flag=1;

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

if(d[j]-d[i]>c\*d2){

j--;

break;

}

if(p[j]<=p[i]) break;

}

if(p[j]<=p[i]){

sum+=((d[j]-d[i])/d2-ct)\*p[i];

ct=0;

}else{

sum+=(c-ct)\*p[i];

ct=c-(d[j]-d[i])/d2;

}

}

if(flag)printf("%.2lf\n",sum);

else printf("No Solution\n");

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

#define MAX\_N 100

float D1,C,D2,P;

float A[MAX\_N],B[MAX\_N];

int N;

void solve()

{

int i;

float res = 0;

float pos,tank;

pos = B[0];

tank = C;

res += pos\*tank;

float we=0;

for(i=1; i<N; i++)

{

tank -= (A[i] - A[i-1]-we)/D2;

if(B[i] < pos)

{

we = 0;

res -= tank\*pos;

tank = C;

pos = B[i];

res += tank\*pos;

}

else

{

if(tank < (A[i+1]-A[i])/D2)

{

we=0;

we = tank\*D2;

tank = C;

pos = B[i];

res += tank\*pos;

}

if(tank < 0)

{

puts("No Solution\n");

return ;

}

}

}

tank -= (D1-A[i-1]-we)/D2;

if(tank < 0)

{

puts("No Solution\n");

return ;

}

res -= tank\*pos;

printf("%.2f\n",res);

}

int main()

{

int i;

scanf("%f%f%f%f%d",&D1,&C,&D2,&P,&N);

A[0] = 0;

B[0] = P;

N++;

A[N] = D1;

for(i=1; i<N; i++)

scanf("%f%f",&A[i],&B[i]);

if(D1 != 0)

solve();

else

puts("No Solution\n");

return 0;

}

本题的Java参考代码如下：

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

double[][] p = new double[1024][1024];

Scanner sc = new Scanner(System.in);

int i, j, k = 0;

double d1 = sc.nextDouble();

double c = sc.nextDouble();

double d2 = sc.nextDouble();

p[0][1] = sc.nextDouble();

int n = sc.nextInt();

n++;

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

p[i][0] = sc.nextDouble();

p[i][1] = sc.nextDouble();

}

p[n++][0] = d1;

double f = c \* d2;

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

if (p[i + 1][0] - p[i][0] > f) {

System.out.println("No Solution");

return;

}

}

double min = 0, max, d;

for (i = 0; i < n - 1; i++) {

d = p[i + 1][0] - p[i][0];

while (d > 0) {

while (p[i + 1][0] - p[k][0] - d >= f)

k++;

for (j = k; j <= i; j++)

if (p[j][1] < p[k][1])

k = j;

max = f - (p[i + 1][0] - p[k][0] - d);

if (max > d)

max = d;

d -= max;

min += max / d2 \* p[k][1];

}

}

System.out.println(String.format("%.2f", min));

}

}