

Заняття 7 (19.10.2016)

Добавил(а) Administrator
29.11.16 09:34 -

\$11. Динамічне програмування . Купування квитків

Ім'я файлу програми:

Bilet.*

Ім'я вхідного файлу:

Bilet.dat

Ім'я вихідного файлу:

Bilet.dat

Максимальний час роботи на одному тесті:

5 секунд

Максимальна оцінка за завдання:

100 балів

За квитками на прем'єру нового мюзиклу вишикувалася черга з N людей, кожний з яких хоче купити 1 квиток. На всю чергу працювала тільки одна каса, тому продаж квитків йшов дуже повільно, приводячи людей черги у відчай. Найкмітливіші швидко відмітили, що, як правило, декілька квитків в одні руки касир продає швидше, ніж коли ці ж квитки продаються поодинці. Тому вони запропонували декільком людям, які стоять підряд віддавати гроші першому з них, щоб він купив квитки на всіх.

Проте для боротьби із спекулянтами касир продавала не більше 3-х квитків в одні руки, тому домовитися таким чином між собою могли лише 2 або 3 підряд вартих людини.

Відомо, що на продаж i -їй людині з черги одного квитка касир витрачає A_i секунд, на продаж двох квитків — B_i секунд, трьох квитків — C_i секунд. Напишіть програму, яка підрахує мінімальний час, за який могли бути продані квитки для всіх людей черги.

Зверніть увагу, що квитки на групу людей, що об'єдналися, завжди купує перший з них. Також ніхто в цілях прискорення не купує зайвих квитків (тобто квитків, які нікому не потрібні).

Формат вхідних даних

У вхідному файлі записано спочатку число N — кількість покупців в черзі ($1 \leq N \leq 5000$). Далі йде N трійок натуральних чисел A_i, B_i, C_i . Кожне з цих чисел не перевищує 3600. Люди в черзі нумеруються починаючи від каси.

Формат вихідних даних

Заняття 7 (19.10.2016)

Добавил(a) Administrator
29.11.16 09:34 -

У вихідний файл виведіть одне число — мінімальний час в секундах, за яке могли бути обслужені всі покупці.

Приклади

Bilet.dat

Bilet.sol

5

5 10 15

2 10 15

5 5 5

20 20 1

20 1 1

12

Заняття 7 (19.10.2016)

Добавил(а) Administrator
29.11.16 09:34 -

2

3 4 5

1 1 1

4

$N=5$

i

Ai

Bi

Ci

1

5

Заняття 7 (19.10.2016)

Добавил(а) Administrator
29.11.16 09:34 -

10

15

2

2

10

15

3

5

5

5

4

Заняття 7 (19.10.2016)

Добавил(а) Administrator
29.11.16 09:34 -

$$D[i] = \min (D[i-1] + A_i, D[i-2] + B_{i-1}, D[i-3] + C_{i-2})$$

Заняття 7 (19.10.2016)

Добавил(а) Administrator
29.11.16 09:34 -

D[4]

D[5]

5

7

5

6

12 – відповідь завдання

$d[0] = 0;$

$d[1] = a[1];$

$d[2] = \text{Мінімальне}(a[1] + a[2], b[1]);$

Для i від 3 до n пц

$d[i] = \text{Мінімальне}(d[i-1] + a[i], \text{Мінімальне}(d[i-2] + b[i-1], d[i-3] + c[i-2]));$

Задача 2. На квадратному аркуші паперу в клітинку розміром $N \times M$ клітинок намальовано кілька прямокутників. Кожен прямокутник складається з цілих клітинок, різні прямокутники накладаються один на одного.

Необхідно написати програму, яка рахує площу покриту цими прямокутниками.

Вхідні дані

В першому рядку N , розмір масиву, в наступних n рядків n масиву, в кожному з яких написані через пробіл n елементів масиву: $A[\text{елемент } I, J] = 1$, якщо клітина $[I, J]$ належить будь-якому прямокутника, і $A[I, J] = 0$, в іншому випадку.

Вихідні дані

Необхідно вивести єдине число - кількість прямокутників.

```
#include <iostream>
```

```
using namespace std;
```

```
int a[100][100];
```

```
int main()
```



```
{  
  
int n,m;  
  
cin>>n>>m;  
  
for (int i=0;i<n;i++)  
  
    for (int j=0;j<m;j++)  
  
        cin>>a[i][j];  
  
int k=0;  
  
for (int i=0;i<n;i++)  
  
    for (int j=0;j<m;j++)  
  
        if (a[i][j]==1)k++;  
  
cout<<k<<endl;
```

```
return 0;
```

```
}
```

Задача 3. "Прямокутники" На квадратному аркуші паперу в клітинку розміром $N \times N$ клітин намальовано кілька прямокутників. Кожен прямокутник складається з цілих клітинок, різні прямокутники не накладаються один на одного і не дотикаються.

Необхідно написати програму, яка рахує число цих прямокутників.

Вхідні дані

В першому рядку N , розмір масиву, в наступних n рядків масиву, в кожному з яких написані через пробіл n елементів масиву: $A[\text{елемент } I, J] = 1$, якщо клітина $[I, J]$ належить будь-якому прямокутника, і $A[I, J] = 0$, в іншому випадку.

Вихідні дані

Необхідно вивести єдине число - кількість прямокутників.

```
#include <fstream>
```

```
using namespace std;
```

```
ifstream cin("input.txt");
```

```
ofstream cout("output.txt");
```

```
int a[100][100];
```

```
int main()
```

```
{
```

```
int n;
```

```
cin>>n;
```

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

```
for (int j=0;j<n;j++)
```

```
cin>>a[i][j];
```

```
int k=0;
```

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

```
for (int j=0;j<n;j++)

if (a[i][j]==1 && a[i+1][j]==0 &&a[i][j+1]==0 && a[i+1][j+1]==0)k++;

cout<<k<<endl;

return 0;

}
```

Задача 4.□**Мишка і зернинки**

В індійському храмі підлогу прямокутної форми вимощено однаковими квадратними плитками 1x1, на кожну з яких насипано від 0 до ***N*** зернинок (***N***<30000). Розміри підлоги ***AxB***

. Мишка вибігає з лівого нижнього кута підлоги храму і рухається до входу у іншу нірку, розміщену у протилежному кутку. Мишка може рухатись лише вправо або вперед, забираючи всі зернинки з клітини, в якій вона знаходиться. Потрібно:

- а) знайти кількість можливих маршрутів руху мишки:
- б) знайти маршрут, рухаючись по якому мишка збере найбільшу кількість зернин.

Вхідний файл **MOUSE.DAT** у першому рядку містить числа **A** та **B** – розміри підлоги ($1 \leq A$

,
B
 $\leq$
100). Далі йде

A
рядків, у кожному з яких розміщено

B
чисел – кількість зернинок на відповідній плитці.

Програма **MOUSE.*** повинна вивести на екран та записати у файл **MOUSE.SOL** у перший рядок кількість можливих маршрутів, у другий рядок – найбільшу кількість зернинок, що може зібрати мишка, у третій рядок – маршрут руху мишки у формі **ППВВВПП**

(
В
– крок вперед,
П
– крок вправо).

Приклад вхідних та вихідних даних:

MOUSE.DAT

MOUSE.SOL

2 3

3

3 2 4

12

1 5 1

ПВП

Заняття 7 (19.10.2016)

Добавил(a) Administrator
29.11.16 09:34 -

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

```
for (int j=1;j<=m;j++)
```

```
a[i][j]=max(a[i][j]+a[i-1][j], a[i][j]+a[i][j-1])
```

3 тур - з 31.10 по 06.11.2016

точка входу для відправлення розв'язків http://134.249.159.199/cgi-bin/new-client?contest_id=35

Задача 1. Мега реклама (20 балів)

Ім'я вхідного файлу: input.txt□□□□□

Ім'я вхідного файлу: output.txt

Ліміт часу: 1с.

На дошці наклеєно декілька листів оголошень. Всі вони прямокутної форми. Деякі листи накладаються частково або повністю. Усі горизонтальні та вертикальні сторони строго взаємо паралельні. Листи, які частково накладаються утворюють многокутник.

Директор рекламного агентства для підрахунку вартості розміщених оголошень наказав менеджерам порахувати загальну суму периметрів усіх утворених таким чином

геометричних фігур (прямокутників, багатокутників). Зрозуміло, що, коли лист із оголошенням повністю перекривається іншим, то периметр першого ніде не враховується.

Вхідні дані

У першому рядку вхідного файлу записано число N - кількість прямокутників. В наступних N рядках записано числа x_1 y_1 x_2 y_2 - декартові координати нижнього лівого та правого верхнього кутів прямокутника. Всі координати - цілі числа що по модулю не перевищують 10000.

Вихідні дані

У вихідний файл потрібно записати суму периметрів утворених геометричних фігур (прямокутників, багатокутників).

Приклад

Приклад вхідних даних

Приклад вихідних даних

7

-15 0 5 10

Заняття 7 (19.10.2016)

Добавил(а) Administrator
29.11.16 09:34 -

-5 8 20 25

15 -4 24 14

0 -6 16 4

2 15 10 22

30 10 36 20

34 0 40 16

228

```
#include "fstream"
```

```
using namespace std;
```

```
ifstream cin("input.txt");
```

```
ofstream cout("output.txt");
```

```
int a[2000][2000];
```



```
int main()
```

```
{
```

```
    int s,k, i,j,n,xz,yz;
```

```
    cin>>n;
```

```
    int *x1=new int[n+1];
```

```
    int *y1=new int[n+1];
```

```
    int *x2=new int[n+1];
```

```
    int *y2=new int[n+1];
```

```
    int maxx=-10000,maxy=-10000,minx=10000,miny=10000;
```

```
    for(k=1; k<=n; k++){
```

```
        cin>>x1[k]>>y1[k]>>x2[k]>>y2[k];
```

```
if (x1[k]<minx) minx=x1[k];
```

```
if (y1[k]<miny) miny=y1[k];
```

```
if (x2[k]<minx) minx=x2[k];
```

```
if (y2[k]<miny) miny=y2[k];
```

```
if (x1[k]>maxx) maxx=x1[k];
```

```
if (y1[k]>maxy) maxy=y1[k];
```

```
if (x2[k]>maxx) maxx=x2[k];
```

```
if (y2[k]>maxy) maxy=y2[k];
```

```
}
```

```
//cout<<minx<<miny<<maxx<<maxy<<endl;
```

```
xz=-minx+1;
```

```
yz=-miny+1;
```

```
minx=minx+xz;
```

```
miny=miny+yz;
```

```
maxx=maxx+xz;
```

```
maxy=maxy+yz;
```

```
//cout<<minx<<miny<<maxx<<maxy<<endl;
```

```
for(k=1; k<=n; k++)
```

```
{
```

```
    x1[k]=x1[k]+xz;
```

```
    x2[k]=x2[k]+xz;
```

```
    y1[k]=y1[k]+yz;
```

```
    y2[k]=y2[k]+yz;
```

```
for(i=x1[k]+1; i<=x2[k]; i++)
```

```
for(j=y1[k]+1; j<=y2[k]; j++)
```

```
{a[i][j]=1;}
```

```
}
```

```
/* for(i=minx;i<=maxx; i++){
```

```
for(j=miny; j<=maxy; j++)
```

```
cout<<a[i][j]; cout<<endl;}
```

```
*/
```

```
s=0;
```

```
for(i=minx;i<=maxx+1; i++)
```

```
for(j=miny; j<=maxy+1; j++){
```

```
if((a[i][j]==0 && a[i+1][j]==1) || (a[i][j]==0 && a[i-1][j]==1) ||
```

```
(a[i][j]==0 && a[i][j+1]==1) || (a[i][j]==0 && a[i][j-1]==1 )) s++;

if((a[i][j]==0 && a[i+1][j]==1 && a[i][j+1]==1) || (a[i][j]==0 && a[i+1][j]==1 && a[i][j-1]==1) ||

(a[i][j]==0 && a[i-1][j]==1 && a[i][j+1]==1 ) || (a[i][j]==0 && a[i-1][j]==1 && a[i][j-1]==1 )) s++;

}

cout<<s<<endl;

return 0;

}
```

Додатково

II етап Всеукраїнської учнівської олімпіади з інформатики (м.Луцьк) 2015-2016н.р. - http://134.249.159.199/cgi-bin/new-client?contest_id=21

Логін school01-2016-school10-2016. Пароль 1.

Заняття 7 (19.10.2016)

Добавил(а) Administrator
29.11.16 09:34 -
