



ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

София, 18 юли 2024 г.

Група С

Задача СТ22. ЛАБИРИНТ

Пояснение към решението

Програмата прочита в масива $c[][]$ дадената във входа двумерна таблица от нули и единици, описваща лабиринта. Прилагаме метода на динамичното програмиране, като запълваме таблицата $t[][][]$. След запълване на таблицата клетката $t[m][i][j]$ съдържа броя на различните пътища, състоящи се от точно m хода, с които се придвижваме от клетката на лабиринта с координати $(1, 1)$ до клетката (i, j) . Очевидно отговорът на задачата се получава в $t[K][N][N]$.

Присвояваме $t[0][1][1]=1$ (има само един път от клетка $(1,1)$ до същата клетка с 0 хода). След това, запълването на $t[][][]$ се извършва последователно в цикъл, в който m се променя от 1 до K . След като е пресметнат броя на всички пътища с $m-1$ до всяка клетка в лабиринта, тръгвайки от началната клетка $(1, 1)$, може да пресметнем броя на всички пътища с m хода чрез двойния цикъл:

```
for(int i=1;i<=N;i++)
for(int j=1;j<=N;j++)
{
    if(c[i][j]=='0')
        t[m][i][j] = t[m-1][i][j-1]+t[m-1][i-1][j]
                    +t[m-1][i][j+1]+t[m-1][i+1][j];
    else t[m][i][j]=0;
}
```

Изполването на примерен масив изисква повече памет, отколкото е дадено в ограниченията на задачата и затова гореописаната програма ще получи само 40% точките. Понеже за пресмятането на стойностите в масива за m се използват само пресметнатите стойности за $m-1$, може да използваме масив, в който първият индекс приема само две стойности: 0 и 1, и променливите $k0$ и $k1$ с начални стойности, съответно 0 и 1, и тези променливи алтернативно да примат стойности 0 и 1 в тялото на цикъла, индексирани от m :

```
for(int i=1;i<=N;i++)
for(int j=1;j<=N;j++)
{
    if(c[i][j]=='0')
        t[k1][i][j] = t[k0][i][j-1]+t[k0][i-1][j]
                    +t[k0][i][j+1]+t[k0][i+1][j];
    else t[k1][i][j]=0;
}
k0=1-k0; k1=1-k1;
```

Накрая отпечатваме

```
if(k1==1) cout<<t[0][N][N]<<endl; else cout<<t[1][N][N]<<endl;
```

Емил Келеведжиев