

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ТАБЛИЦА

Решение 1: Тръгваме отляво-надясно в таблицата с размери m реда и n стълба и намираме сумата от елементите на всяка подтаблица с размери u реда и v стълба. През това време търсим най-маалката по-абсолютна стойност сума. За целта се използват четири вложени цикли.

```
int r=999999999;
for(int i=0;i<=m-u;i++)
for(int j=0;j<=n-v;j++)
{
    int s=0;
    for(int i1=0;i1<u;i1++)
    for(int j1=0;j1<v;j1++)
        s += a[i+i1][j+j1];
    if(abs(s)<r) r=abs(s);
}

printf("%d\n",r);
```

Този алгоритъм работи за 60% от тестовите примери.

Решение 2: Тръгваме отляво-надясно по редовете на таблицата A с размери m реда и n стълба и намираме последователно сумите от v на брой елементи на всеки ред. Записваме получените суми в друга таблица AX като първият ред съдържа всички получени суми от по v елемента от първия ред на таблица A , вторият – всички суми от по v елемента от втория ред на таблица A и т.н.

```
for(int i=0;i<m;i++)
{
    int s=0;
    for(int j=0;j<v;j++) s+=a[i][j];
    ax[i][0]=s;
    for(int j=v;j<n;j++)
    { s = s + a[i][j] - a[i][j-v];
      ax[i][j-v+1]=s;
    }
}
```

За да намерим сумите на елементите на подтаблица с размери u реда и v стълба, в едномерен масив B пресмятаме и записваме сумите от по u елемента от всеки стълб на таблица AX , тръгвайки по редове.

```
for(int j=0;j<=n-v;j++)
    for(int i=0;i<u;i++) b[j] += ax[i][j];
```

Намираме най-малката по абсолютна стойност на елементите на масива B .

```
int r=999999999;
```

```
for(int j=0;j<=n-v;j++)
{ int s=b[j];
  if(abs(s)<r) r=abs(s);

  for(int i=u;i<m;i++)
  {
    s = s + ax[i][j]-ax[i-u][j];
    if(abs(s)<r) r=abs(s);
  }
}
```

Този алгоритъм решава напълно задачата. Както се вижда от приложените фрагменти на програмата, за реализирането му са използвани два вложени цикли.

Автор: Зорница Дженкова