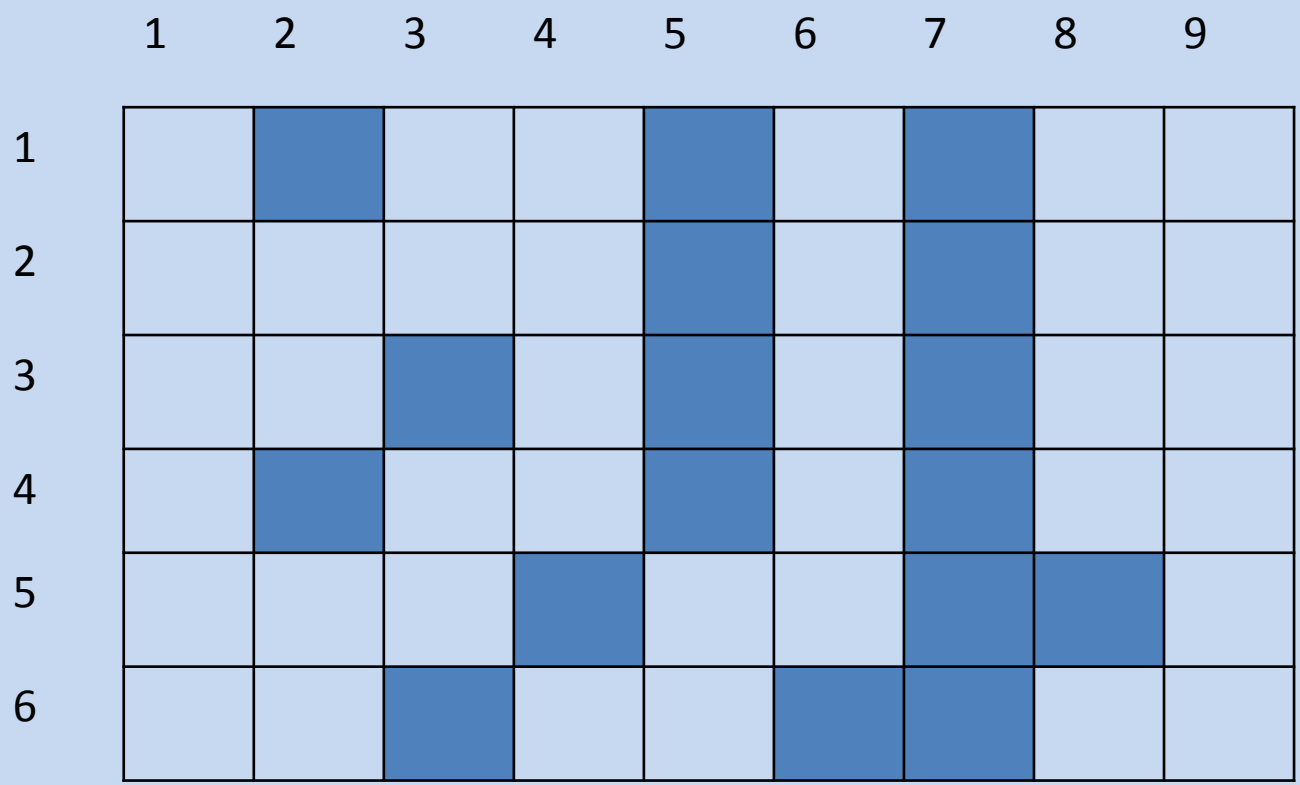
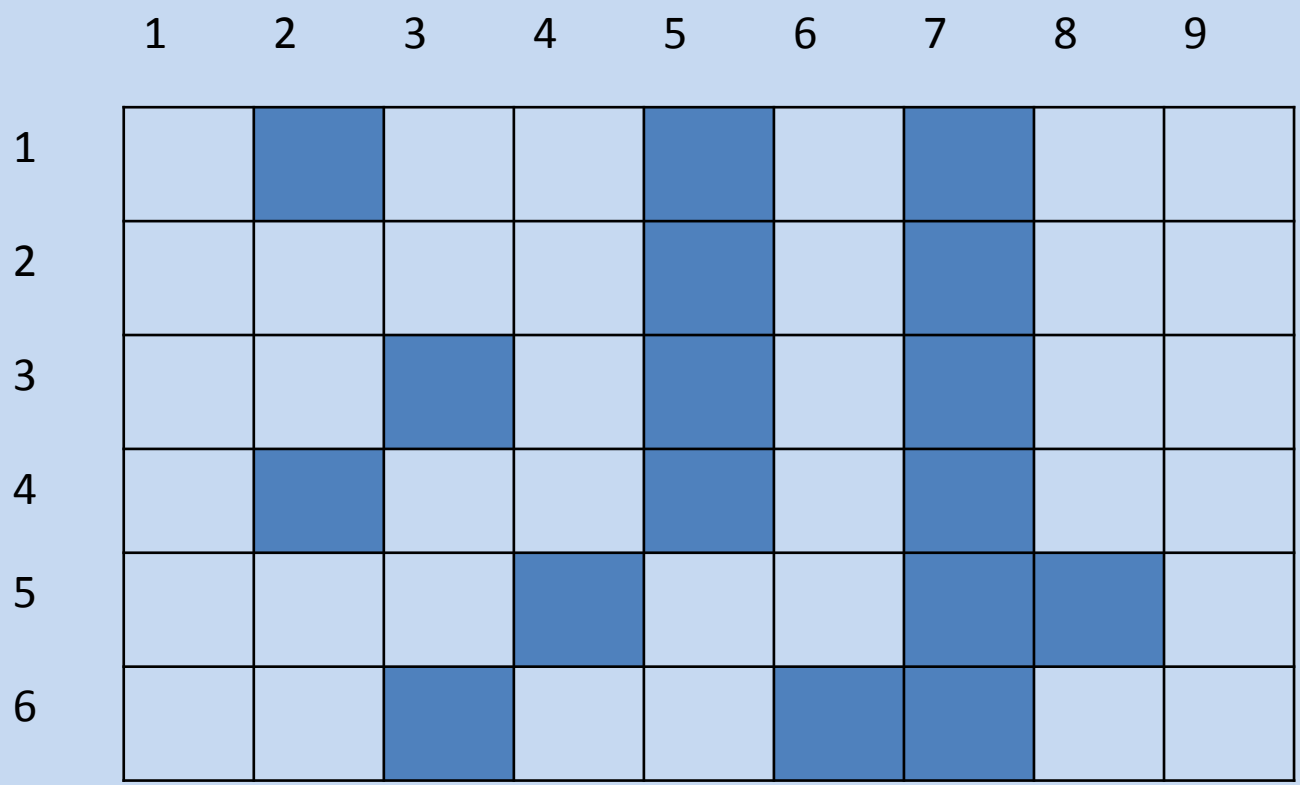
Лабиринти

Поле от празни и пълни клетки



Преминаване от клетка в съседна (според дефиницията за съседност). Свързани области.



Алгоритъм "Вълна":
Запълване на празните клетки от лабиринта с последователни слоеве.
(разпространяване в ширина, търсене в ширина)

	1	2	3	4	5	6	7	8	9
1	1		3	4		10			
2	2	2	3	4		9			
3	3	3		4		8			
4	4		4	5		7			
5	5	5	5		6	7			
6	6	6		6	7				

Програма за реализация

```
const int M=21, N=21;
int a[M][N];
int m,n;

int di[]={1,1, 1,0, 0,-1,-1,-1};
int dj[]={1,0,-1,1,-1, 1, 0,-1};
```

Четене на входните данни

```
cin >> m >> n;
  int k;
  cin >> k;
  for(int t=1;t<=k;t++)
  {
    int i,j;  cin >> i >> j;
    a[i][j]=-1;
  }
```

Запълване на слой

```
bool spread(int c)
{ bool b=false;
  for(int i=1;i<=m;i++)
  for(int j=1;j<=n;j++)
  if(a[i][j]==0)
  { for(int t=0;t<=7;t++)
     {  int i0=i+di[t]; int j0=j+dj[t];
        if(i0>0 && i0<=m && j0>0 && j0<=n)
         if(a[i0][j0]==c-1) {a[i][j]=c; b=true;}
     }
  }
  return b;
}
```

Цялата вълна

```
void wave()
{
  a[1][1]=1;
  int c=1;
  for(int i=1;i<=m;i++)
  for(int j=1;j<=n;j++)
  {
    c++; if(!spread(c)) return;
  }
}
```

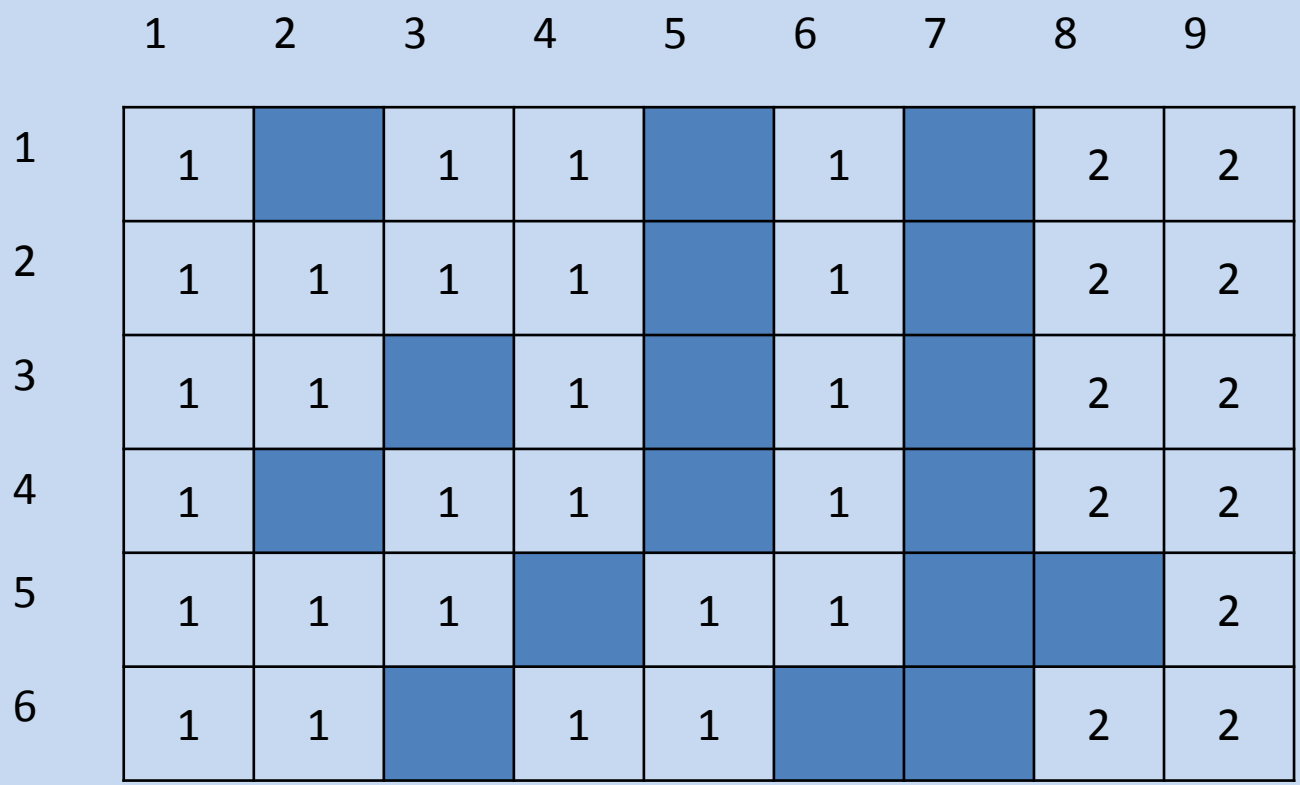
Приложения на алгоритъма "Вълна"

- Намиране на най-къс път
  - Дължина на най-късия път
  - Описание на най-късия път
- Намиране на областите на свързаност
  - Брой на областите на свързаност
  - Брой на клетките във всяка от областите на свързаност

Алгоритъм "Вълна":
Най-къс път от клетка (1,1) до клетка (1,6)

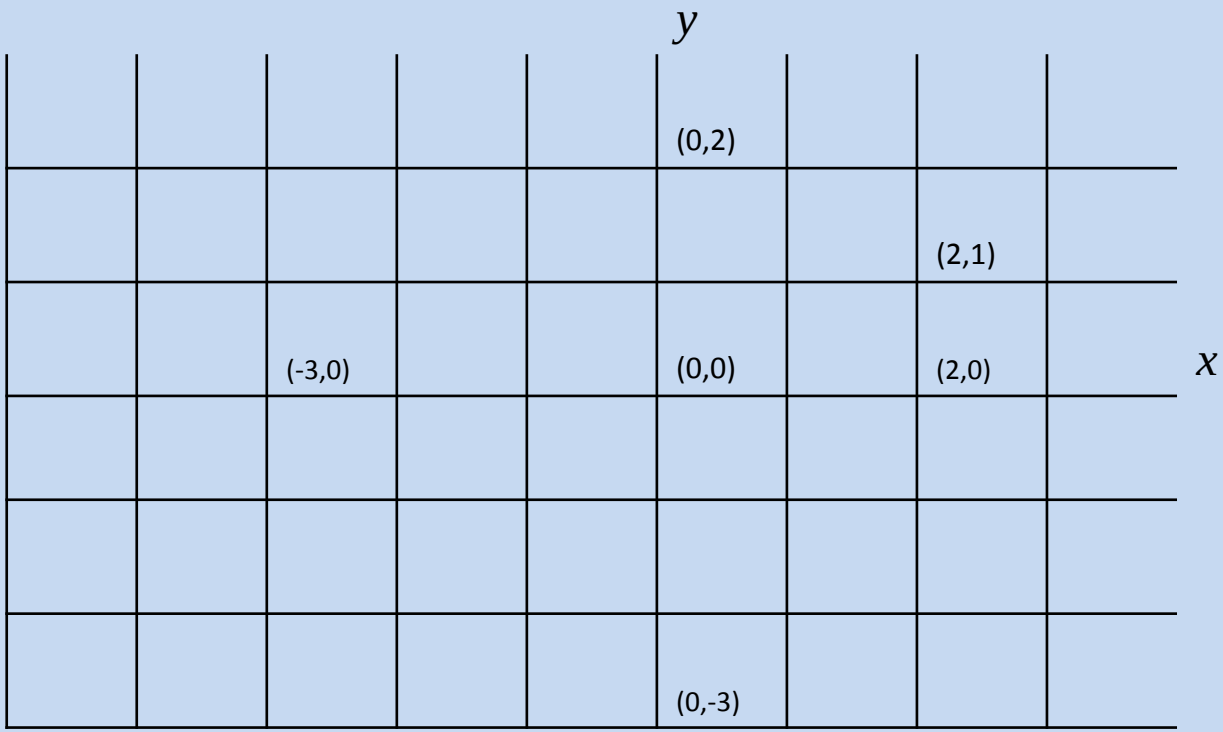


Алгоритъм "Вълна":
Намиране на области на свързаност

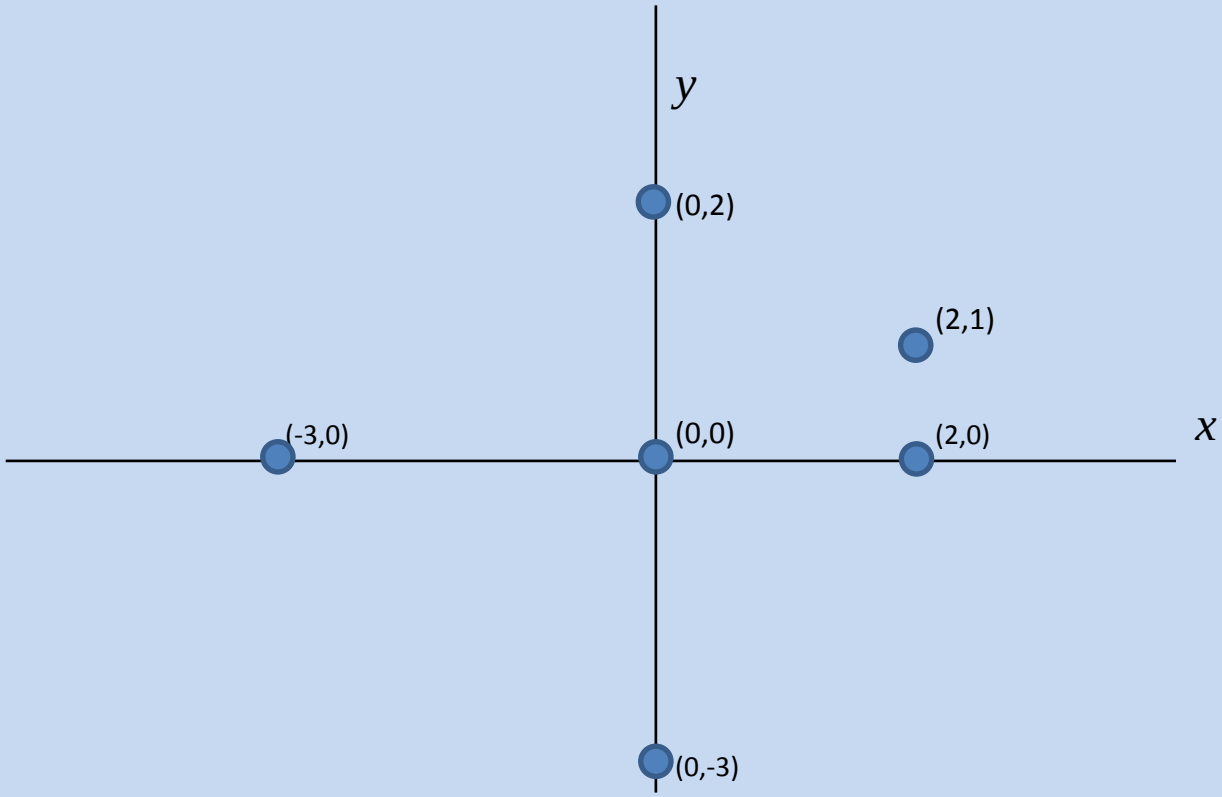


Координати на точки в равнината

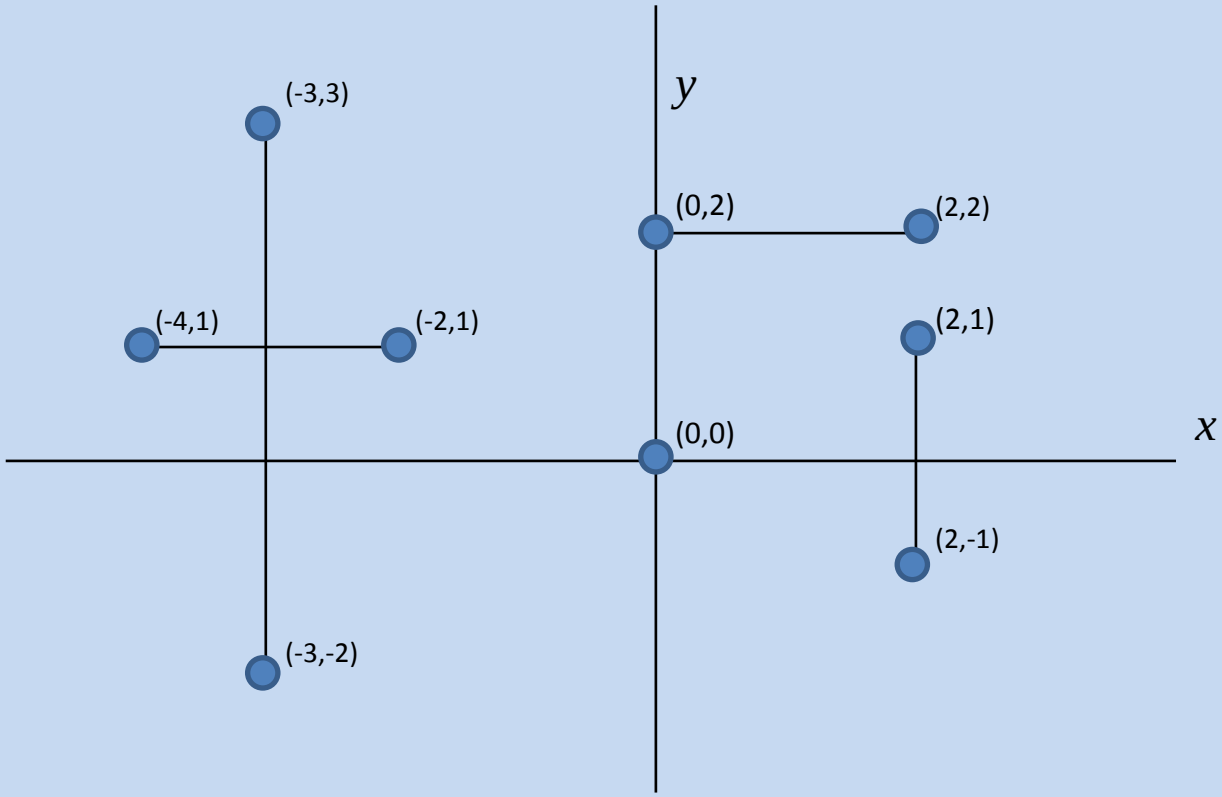
(прилики и разлики с координати на клетка в поле от клетки)



Координати на точки в равнината
(точки с целочислени координати)



Хоризонтални и вертикални отсечки с целочислени координати на крайните им точки



Намиране на пресечни точки с координатните оси

Отсечката AB, задаваме с координатите на крайните ѝ точки:

ax, ay, bx, by

Отсечката е вертикална, ако ax==bx

Отсечката е хоризонтална, ако ay==by

Пресечната точка с абсцисата на правата, определена от вертикална отсечка има координати: ax, 0

Пресечната точка с ординатата на правата, определена от хоризонтална отсечка има координати: 0, bx

Дали точка лежи върху отсечка

Точка P с координати: x, y

Отсечка AB с координати: ax, ay, bx, by

Случай, когато отсечката е хоризонтална,

т.е. ay=by

Осигуряваме ax < bx и прилагаме проверка чрез

If (y==ay && ax <= x && x <= bx) ....

Няколко отсечки

Използване на масив за записване на координатите на крайните им точки:

Първият край на i-тата отсечка има координати: x[1][i] и y[1][i]

Вторият край на i-тата отсечка има координати: x[2][i] и y[2][i]

i=1, 2, 3, ..., n.

Пресичане на две отсечки

Хоризонтална отсечка (с номер i)
и вертикална отсечка (с номер j):

За отсечката с номер i двата края имат координати:

(x[1][i], y[1][i]) и (x[2][i], y[2][i]), като y[1][i]==y[2][i]

За отсечката с номер j двата края имат координати:

(x[1][j], y[1][j]) и (x[2][j], y[2][j]), като x[1][j]==x[2][j]

Осигуряваме x[1][i] < x[2][i] и y[1][j] < y[2][j] и

правим проверката чрез:

```
if(       x[1][i]<=x[1][j] && x[1][j]<= x[2][i]
    && y[1][j]<=y[1][i] && y[1][i]<= y[2][j] ) ...
```

Правогълници със страни, успоредни на координатните оси

В много от задачите те са задават чрез координатите на две свои противоположни точки, т.е. чрез крайните точки на един от диагоналите си

ax, ay и bx, by

При въвеждане на данните е добре да осигурим да бъде ax < bx и ay < by

Задачи при правогълници със страни, успоредни на координатните оси

Проверка дали точка P с координати x,y е вътрешна или е контурна за правоъгълника:

if(ax <= x && x <= bx && ay <= y && y <= by) ...

Проверка дали два правоъгълника имат обща вътрешна или обща контурна точка – свежда се до разглежданите задачи за точки и отсечки.