Announcements

- Buddy groups have been formed
  - Find it in classroom Study Groups - Google Sheets
  - Reach out to your buddies for contact info in Zoom chat now
  - Start to check with your buddies for class/homework/questions or have group study, etc.
- Use your real name in XJOI and classroom
- If you can't complete 100% of homework last week, don't be upset. We will review it in detail today. Please finish your missing parts after class (important!)

Algorithm Video Competition

- Algorithm video competition.pptx - Google Slides
- Registration Form: https://forms.gle/WTJfmzTgJxyJ2nw29
- Not mandatory but highly recommended
- Good prizes and bonus points for midterm exam
- Great opportunity to learn, showcase creativity and practice presentation skills

Homework last week (P9561)

Friendship Circle

Given a n by n matrix M, representing the friendships in class. If M[i][j] = 1, i-th student and j-th student are friends to each other. Please compute the number of social circles in total.

Sample Input:

```
3

1 1 0

1 1 1

0 1 1
```

Sample Output:

```
1
```

Homework last week (P9561)

```
8  void findFriend(int i) // dfs search
9  {
10     if (maze[i][i] != 1) // this row was checked
11         return;
12     maze[i][i] = 2; // mark this row as checked
13     for(int j = 0; j < n; j++)
14         if (maze[i][j] == 1)
15             findFriend(j); // check new row
16 }
17
```

```
23
24     for(int i = 0; i < n; i++)
25         if (maze[i][i] == 1) { // new group
26             group++;
27             findFriend(i);
28         }
29
```

Homework last week (P8100)

Get water

> n people are lining in front of a tap to get some water. Suppose it takes Ti time for each person to get water. Please design a program to find out the sequence of those people in the line, so the average waiting time for those n people would be minimal.

Sample input 1 :

10

56 12 1 99 1000 234 33 55 99 812

Sample output 1 :

3 2 7 8 1 4 9 6 10 5

291.90

Homework last week (P8100)

```cpp
#include <bits/stdc++.h>
using namespace std;

struct water { // define struct for time and index
    int time, index;
    water(int x, int y) : time(x), index(y) {};
};
bool comp(water a, water b) // need this for sorting
{
    if (a.time != b.time)
        return a.time < b.time;
    return a.index < b.index;
}

```

```cpp
    sort(vec.begin(), vec.end(), comp);
    double total = 0;
    for(int i = 0; i < n-1; i++)
        total += (n-i-1) * vec[i].time; // total waiting time
    double d = total/n; // average waiting time
    for(int i = 0; i < n; i++)
        printf("%d ", vec[i].index+1);
    printf("\n%.2f", d);
```

Homework last week (P9397)

Give a Row × Rol capital letter matrix, the starting position is the upper left corner. You can move up, down, left and right, and can't move to the letters that have passed. The question is how many letters you can go through at most.

[data format]
In the first row, enter the number of rows R and columns S of the alphabetic matrix (1 ≤ R, S ≤ 20). Then enter the letter matrix in row R and column S.

Output the maximum number of different letters that can be passed.

Sample input:

```
3 6
HFDFFB
AJHGDH
DGAGEH
```

Sample output:

```
6
```

Sample input:

```
3 6
HFDFFB
AJHGDH
DGAGEH
```

Homework last week (P9397)

```
5  char a[23][23]; // max is 20x20
6  int dx[4]={0,1,0,-1};
7  int dy[4]={1,0,-1,0};
8  bool f[135]; // visited letters, Z is 132
9
10 void dfs(int x,int y,int step)
11 {
12   if(step>ans) ans=step;
13   for(int i=0;i<4;i++){
14     int tx=dx[i]+x;
15     int ty=dy[i]+y;
16     if(tx>=1&&ty>=1&&tx<=r&&ty<=s&&!f[a[tx][ty]]){
17       f[a[tx][ty]]=1;
18       dfs(tx,ty,step+1);
19       f[a[tx][ty]]=0; // back tracking!
20     }
21   }
22 }
```

```
29   f[a[1][1]]=1; // start from upper left corner
30   dfs(1,1,1);
31   cout<<ans;
32   return 0;
```

Sample input:

```
3 6
HFDFFB
AJHGDH
DGAGEH
```

Homework

1

Review

Student spends time reviewing class materials to retain information

2

Self Study

Step 1:Pencil/paper

Step 2: Coding

Step 3: Debugging

3

Study Buddy

If stuck for more than 30 mins with no code, reach out to buddy for help

4

Office Hours

Get deeper insight and clear ideas from experienced TAs help

X - Camp

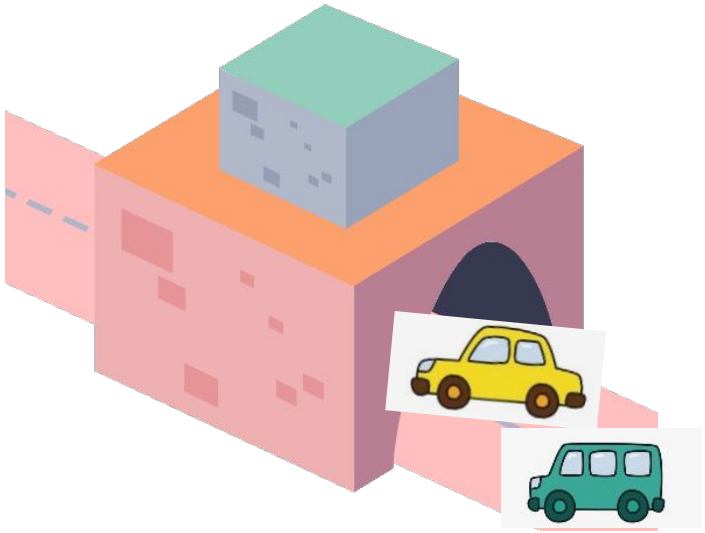
Queue

Arrays simulate queue and stl::queue

In this course we will learn a new data structure

Class : 202

Scenario import (FIFO)



1. A tunnel with only one channel

2. Window that does not allow queue cuts

Can you give a similar example?

Data structure model

When using a queue to access data, data can only enter the queue from one end of the queue (tail), and exit the queue from the other end (head).

Queue functions

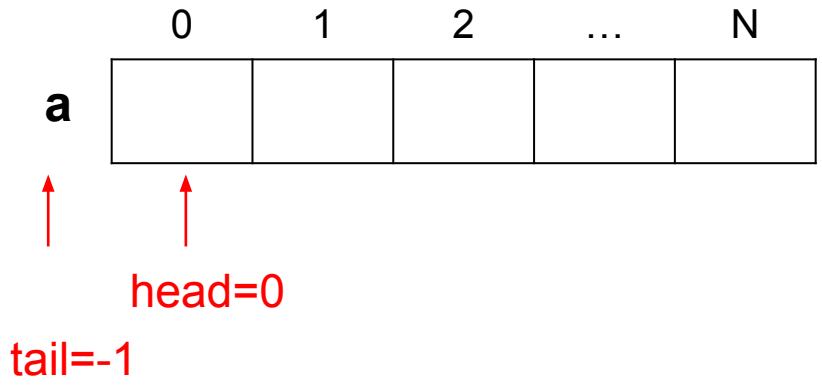
- Empty
- Size
- Front
- Back
- Push (enqueue)
- Pop (dequeue)

If we need to build a queue, what data structure can we use?

Build queue

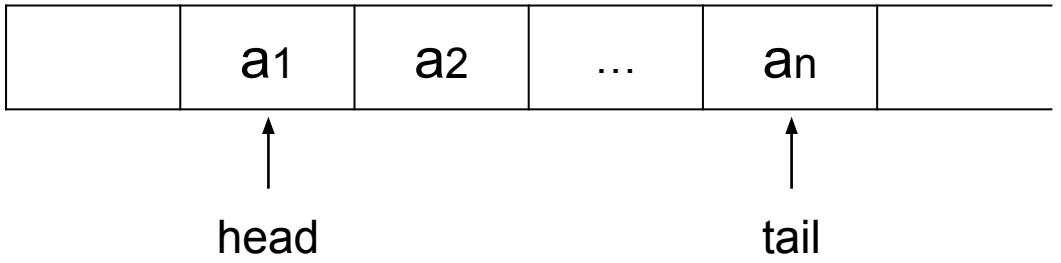
We need :

- A array : **■int que[N]**
- Two variables : **■int head , tail**
- initialization : **■head=0 , tail=-1**
- enqueue : **■que[++tail]=data**
- dequeue : **■head++**



Some methods of queue

- Query queue head data : ■**a [head]**
- Query queue tail data : ■**a [tail]**
- The length of the queue : ■**tail-head+1**
- Queue is empty? : ■**head>tail**



Questions 1

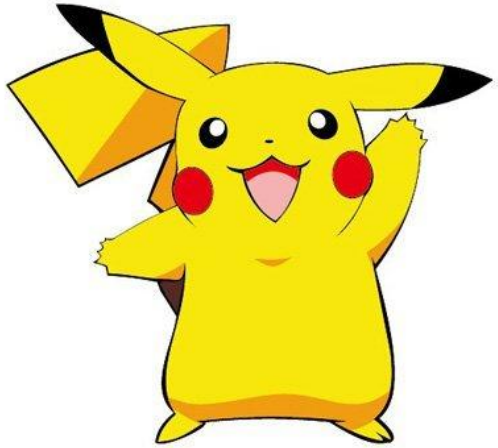
● Add data a, b, c, d to the empty queue, followed by two delete operations, the head data at this time is ( )

A. d

B. b

C. c

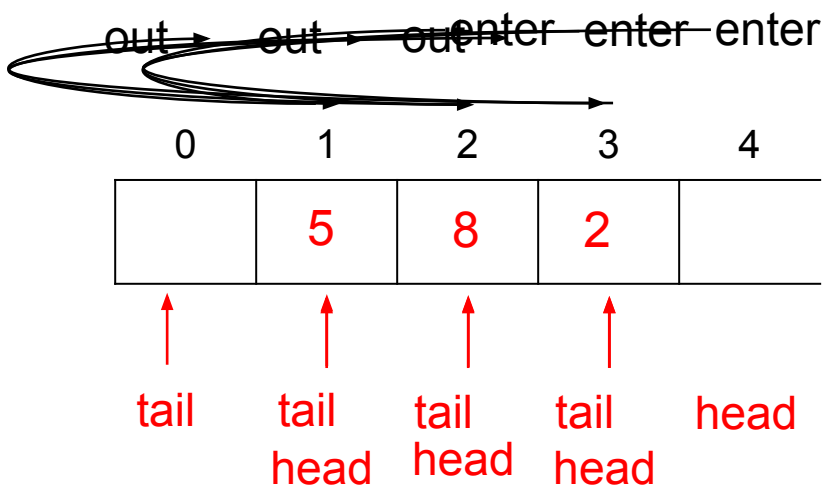
D. a



Queue functions

enqueue

```
void push(int x){
        a[++tail];
}
```



push-> pop -> push->pop->pop

dequeue:

```
void pop(){
        head++ ;
}
```

size:

```
int size(){
        return tail-head+1;
}
```

Queue is empty?

```
bool empty(){
        return head>tail;
}
```

std::queue

head file : #include<queue>

queue definition : queue<int> que;

related functions :

back()	return the last element
empty()	Returns true if the queue is empty
front()	return the first element
pop()	remove the first element
push()	add an element at the end
size()	Returns the number of elements in the queue

STL Queue Example

```
#include<queue>
...
  queue<int> q;
  q.push(0);
  q.push(3);
  cout << q.front();     //?
  q.pop();
  cout << q.front();     //?
  cout << q.back();    //?
  cout << q.empty();  //?
  cout << q.size();      //?
```

Struct

X-Camp

Struct - composite structure of data

Vs. primitive data type: int, string, char, bool, ...
Sometimes something might have multiple properties

For example:
Student: name, age, gender, grade
Point: x, y
Score: math, language, science

Is there a way to express it?

Struct - composite structure of data

Yes, of course, C++ is powerful

```
struct student {
  string name;
  int age;
  char gender;
  int grade;
};
```

```
struct point {
  Int x;
  int y;
};
```

```
struct score {
  string name;
  string id;
  Int math;
  int language;
  Int science;
};
```

Define variables with struct

After you define a struct, then you can use it to declare variables, just like using the primitive data type

```
struct student {
  string name;
  int age;
  char gender;
  int grade;
};
```

```
student amy;

student xcamp[10];

queue<student> sq;
```

Struct initialization and value assigment

After you define a struct, then you can use it to declare variables, just like using the primitive data type

```
struct student {
  string name;
  int age;
  char gender;
  int grade;
};
```

```
student amy = {
      “Amy Wang”,
      10,
      ‘F’,
      6
}
```

```
student amy;
amy.name =  “Amy Wang”,
amy.age = 10;
amy.gender = ‘F’;
amy.grade = 6;

student s = amy;
```

Struct - access value in structure

```
struct student {
  string name;
  int age;
  char gender;
  int grade;
};
```

```
student amy = {
      “Amy Wang”,
      10,
      ‘F’,
      6
}
```

```
cout << amy.name;
cin >> amy.name;
int a = amy.age;
char gender = amy.gender;
Int gap = 18 - amy.age;
```

Struct - constructor

```
struct point {
    int x, y;
    point(int a, int b) : x(a), y(b) {}
};

struct point {
    int x, y;
    point(int a, int b) {
        x = a;
        y = b;
    }
};
```

Constructor allows you to do

```
point p(10, 20);
```

Equals to:

```
point p;
p.x = 10;
p.y = 20;
```

```
q.push(point(x, y));
```

Equals to:

```
point p;
p.x = 10;
p.y = 20;
q.push(p);
```

Queue practice

- Define a struct STUDENT, with name, ID, test score
- Define a queue of STUDENT
- Push 3 students into queue
- Take out a student from front (remove)
- Check if queue is empty
- Print out size of queue

X - Camp

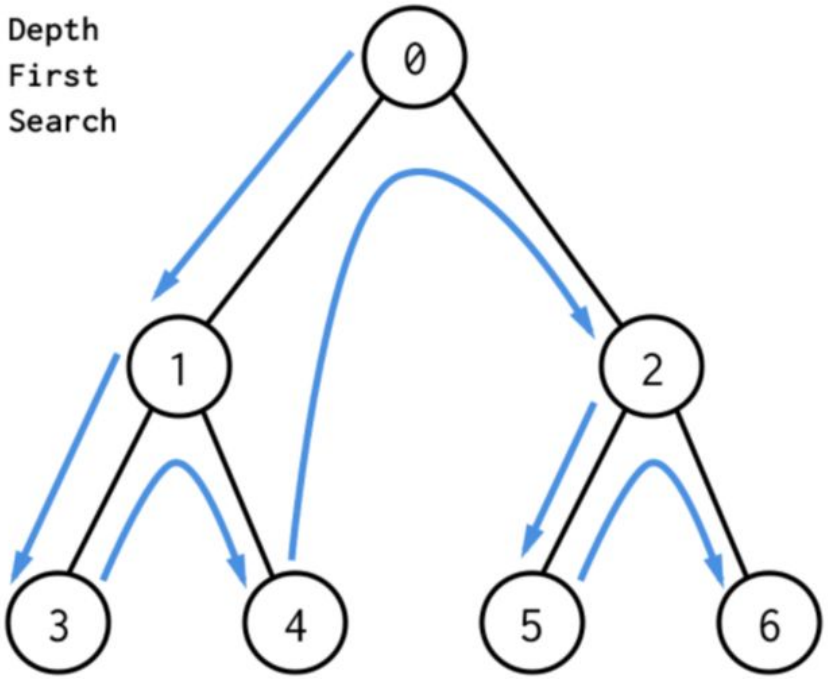
Breadth First Search

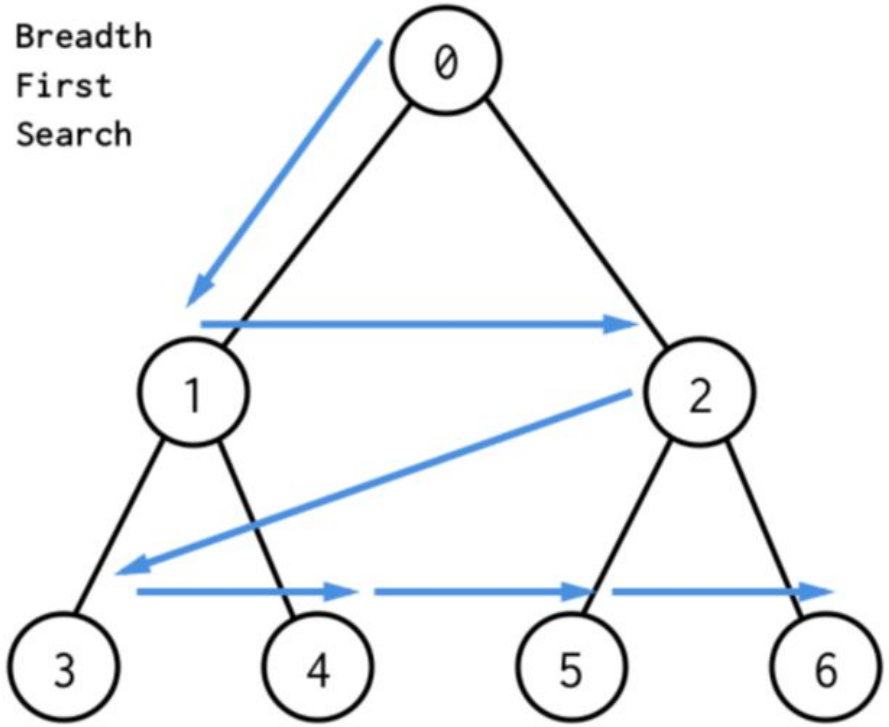
Search algorithm (BFS), implemented using queue

In this course, we will learn a new algorithm

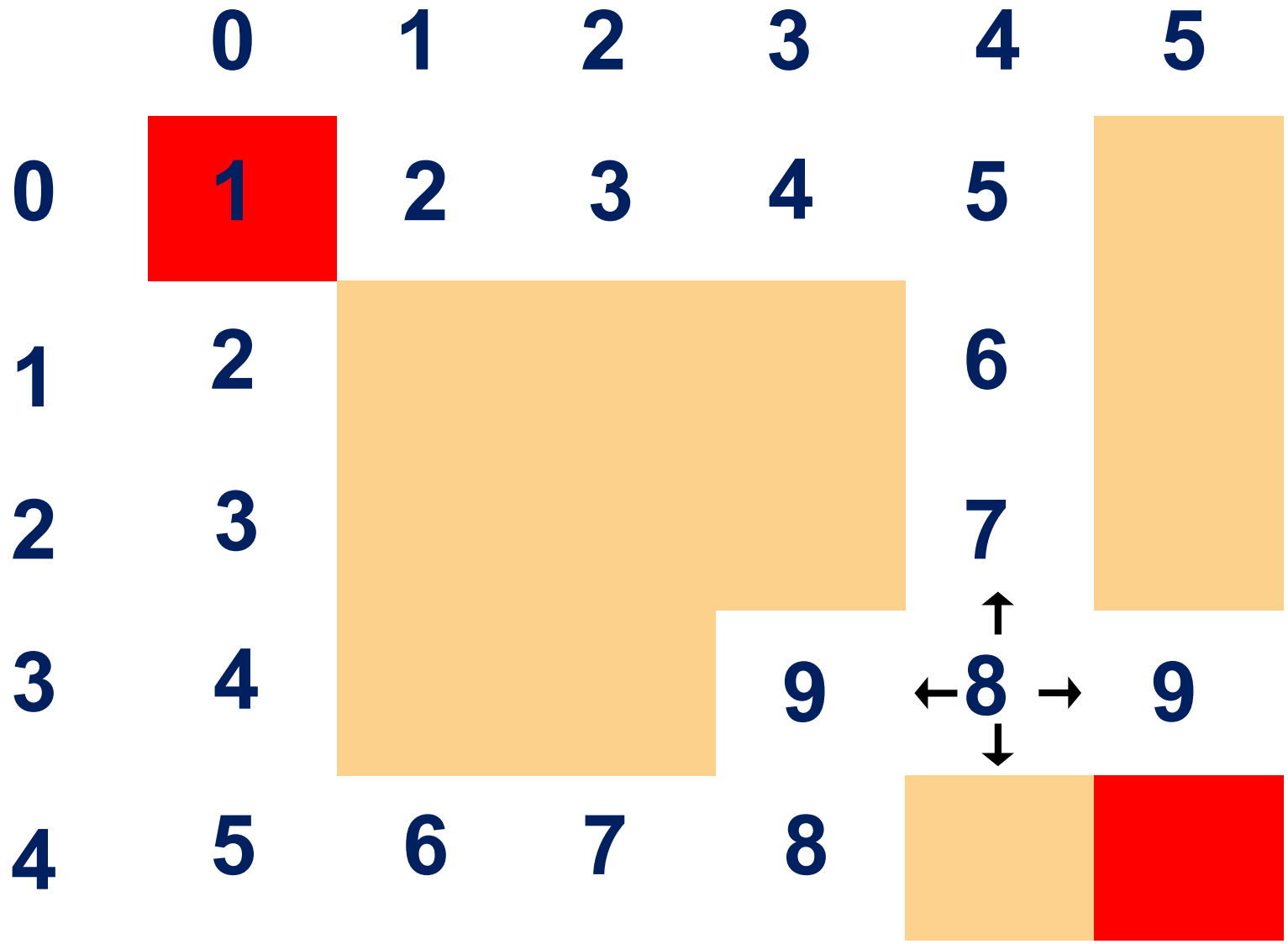
Class : 202

DFS vs BFS





Use BFS to walk the maze



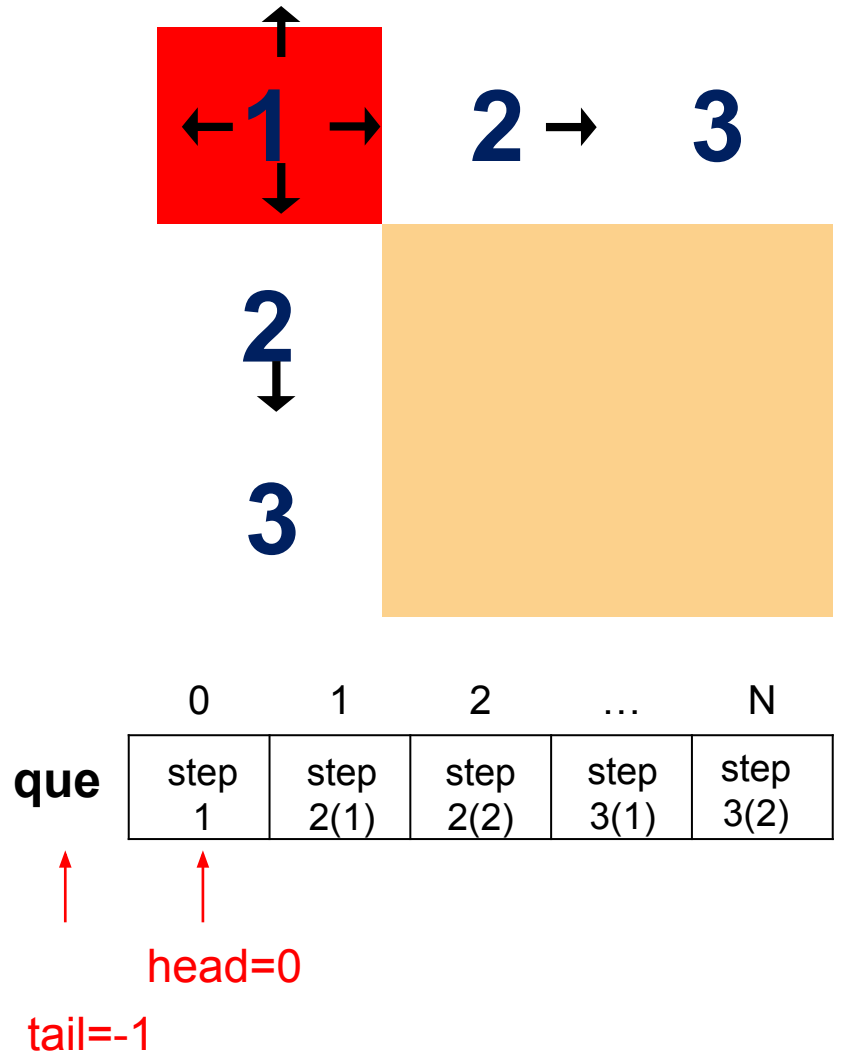
BFS

BFS walks the maze:

Like a team starting from a starting point, there are the following rules:

- 1. Encounter a branch to split the team
- 2. Synchronized forward

Synchronous advance can use queue
as the picture shows：



BFS - find distance of the shortest path

Data structure - what do you need?

- maze - a 2D array of char or int — **int maze[N][N]**
- visited - a 2D array of bool — **bool visited[N][N]**
- distance of each point - a 2D array of unsigned int — **unsigned dist[N][N]**
- struct node - (int x, y) — **struct node {int x, y;}**
- node queue - the queue of the current bases to expand from — **queue<node> q;**
- m expanding directions — **int dx[m], dy[m];**

BFS - how to do it bfs (sx, sy)

Procedure - how to do it bfs (sx, sy)

1. Add the start node to the queue
2. While queue is not empty
   a. Take the front node out of the queue
   b. Visit all next nodes from this node
      i. Set the distance (current dist + 1)
      ii. If it's the end node, done.
      iii. Mark visited true
      iv. Push the new node to the queue

```
void bfs(sx,sy){
    q.push(sx,sy);
    while(!q.empty()){
        (x,y) = q.front();
        for(all the directions){
            nx = x + dx[i];
            ny = t + dy[i];
            if(new location now valid)
        continue;
            dist[nx][ny] = dist[x][y] + 1;
            if( (nx,ny) is end point ){ done. };
            visited[nx][ny] = 1;
            q.push( point(nx,ny) );
        }
    }
}
```

BFS Practice

https://onlinegdb.com/HxPkXCiZu

```
int maze[5][5] =   {{0,0,0,0,0 },
                   {0,1,1,1,0 },
                   {0,1,0,0,0 },
                   {0,1,0,1,0 },
                   {0,0,0,1,0 } };
```

Find shortest distance from (0, 0) to (4, 4). It should be 8.

Sample code

https://onlinegdb.com/4swkhK8Ie

```
int maze[5][5] =   {{0,0,0,0,0 },
                   {0,1,1,1,0 },
                   {0,1,0,0,0 },
                   {0,1,0,1,0 },
                   {0,0,0,1,0 } };
```

Find shortest distance from (0, 0) to (4, 4)

Homework this week (P3564)

Print scores

Print out all scores below average among n scores, 1<=n<=100.

Sample Input

3 40 50 60

2 90 80

5 10 10 90 80 80

Sample Output

40

80

10 10

1. How to read input continually?
2. How to calculate average score?

Homework this week (P8120)

Red and black

A rectangular room is covered by square tiles which is either red or black.

A man stands on a black tile, and he can move to one of the adjacent four tiles. The rule is he can only move to the black tiles, but not the red ones.

Write a program to calculate the number of black tiles he can reach.

https://xjoi.net/contest/4382 then go to problem 2

Use BFS

How to check if a neighbor is available?

Print out maze after reading to ensure it is correct.

Homework this week (P8110)

There is a n×n maze which only has number 0 and 1. If you are in a spot 0, then you can move to one of the four adjacent cells with 1 in it. If you are in 1, then you can move to one of the four adjacent cells with 0 in it. Given the maze, find out how many cells you can move into (including the original cell).

Input:

The first line contains two positive integers n, m.

The next n lines, each line contains n characters (0 or 1, with no space in between)

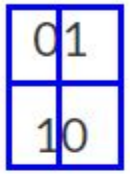
The next m lines, each line contains two positive integers i and j (separated by space), which means the cell in the i-th row and the j-th column (the starting point).

Output:

m lines, output the answer for each inquiry.

sample input 1:

2 2



1 1

2 2

sample output 1:

4

4

Note:

For 100% data, $n \le 1000, m \le 100000$。

Floodfill optimization for multiple inputs

If the number of connected blocks in the same area is queried multiple times

Calculate again? 

Save the results for calculated area.

```
//a global variable to save results
int results[area_id];
if (results[area_id] > 0) return results[area_id];
```

sample input 1:

```
2 2
01
10
1 1
2 2
```

sample output 1:

```
4
4
```

Note:

For 100% data, $n \le 1000, m \le 100000$。

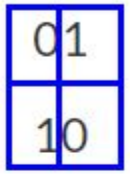
Homework this week (P8110)

Possible solution:

- Read input (try your own way but if you have trouble, there is hint on homework page)
- Use another array to store area ID
- Use result[area_id] to store filled cells for this area
- For a new query, if cell has been filled, get existing answer

sample input 1:

2 2



1 1

2 2

sample output 1:

4

4

Note:

For 100% data，$n \le 1000, m \le 100000$。