bfs

人员

田心一、李瑞涵、纪博涵、蒋叔璋、初锦阳、苑钊、高健桓、刘宸熙 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/2022

比赛概况 | 题目列表 | 选择题列表 | 提交记录 | ★ 实时榜单 | ★ 选择题排行榜

王向东老师周日三点半C++搜索剪枝

刷新

#	用户名	姓名	编程分	时间	A	B	C	D	E
1	liuliwei	柳力玮	400	189	100	100	100	100	
2	liruihan	李瑞涵	350	239	100	100	100	50	
3	tianxinyi	田心一	300	152	100	100	100		
4	yuanzhao	苑钊	300	204	100	100	100		0
5	lizhishuo	李知朔	300	242	100	100	100		
6	zhaoshufan	赵书梵	300	255	100	100	100		
7	gaojianhuan	高健桓	290	279	100	100	90		
8	jiangshuzhang	蒋叔璋	200	147	100	100	0		
9	jibohan	纪博涵	200	148	100	100	0		
10	chujinyang	初锦阳	200	153	100	100	0		

本周作业

https://cppoj.kids123code.com/contest/2172 (课上讲了 A ~ E 题, 课后作业是 F 题)

课堂表现

今天讲了 bfs 这个知识点, 这个知识点不难, 同学们课上听讲都听的很认真。

课堂内容

P2040 打开所有的灯

每个灯如果重复两次的话, 等于没操作

所有每个灯只有 动一次 或者 没动 两种情况, 因此可以 2^n 枚举所有可能即可。

```cpp
#include <bits/stdc++.h>

using namespace std;

const int maxn = 5 + 5;
int w[maxn][maxn], a[maxn][maxn];
bool st[maxn][maxn];
```

```cpp
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};

bool check() {
  for (int i = 0; i < 3; ++i) {
    for (int j = 0; j < 3; ++j) a[i][j] = w[i][j];
  }

  for (int i = 0; i < 3; ++i) {
    for (int j = 0; j < 3; ++j) {
      if (st[i][j]) {
        a[i][j] = 1 - a[i][j];
        for (int k = 0; k < 4; ++k) {
          int ni = i+dx[k], nj = j+dy[k];
          if (ni>=0 && ni<3 && nj>=0 && nj<3) a[ni][nj] = 1 - a[ni][nj];
        }
      }
    }
  }

  for (int i = 0; i < 3; ++i) {
    for (int j = 0; j < 3; ++j) {
      if (!a[i][j]) return false;
    }
  }
  return true;
}

int main()
{
  for (int i = 0; i < 3; ++i) {
    for (int j = 0; j < 3; ++j) cin >> w[i][j];
  }

  int res = 10000000;
  for (int i = 0; i < (1<<9); ++i) {
    int cnt = 0;
    for (int j = 0; j < 9; ++j) {
      int x = j/3, y = j%3; st[x][y] = (i>>j)%2;
      if ((i>>j)%2 == 1) ++cnt;
    }
    if (check()) res = min(res, cnt);
  }
  cout << res << endl;
  return 0;
}
```

离开中山路

bfs 模板, dis[x][y]: 维护到 (x,y) 这个点的最短路是多少

```cpp
#include <bits/stdc++.h>

using namespace std;

const int maxn = 1000 + 5;
char s[maxn][maxn];
int dis[maxn][maxn];
struct node {
  int x, y;
};
int dx[] = {-1, 1, 0, 0};
int dy[] = {0, 0, -1, 1};

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> (s[i]+1);
  int sx, sy, ex, ey; cin >> sx >> sy >> ex >> ey;

  memset(dis, -1, sizeof(dis));
  queue<node> q; q.push({sx, sy}); dis[sx][sy] = 0;

  while (!q.empty()) {
    node u = q.front(); q.pop();
    int x = u.x, y = u.y;
    for (int i = 0; i < 4; ++i) {
      int nx = x+dx[i], ny = y+dy[i];
      if (nx>=1 && nx<=n && ny>=1 && ny<=n && s[nx][ny]!='1' && dis[nx][ny]==-1) {
        q.push({nx, ny}); dis[nx][ny] = dis[x][y] + 1;
      }
    }
  }

  cout << dis[ex][ey] << endl;
  return 0;
}
```

[JOI2022 预选赛 R2] 地毯 / Carpet

做法跟上一道题一样, 只有判断条件需要改一下

```cpp
#include <bits/stdc++.h>

using namespace std;

const int maxn = 500 + 5;
char s[maxn][maxn];
int dis[maxn][maxn];
struct node {
  int x, y;
```

```
};
int dx[] = {-1, 1, 0, 0};
int dy[] = {0, 0, -1, 1};

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> (s[i]+1);

  memset(dis, -1, sizeof(dis));
  queue<node> q; q.push({1, 1}); dis[1][1] = 0;

  while (!q.empty()) {
    node u = q.front(); q.pop();
    int x = u.x, y = u.y;
    for (int i = 0; i < 4; ++i) {
      int nx = x+dx[i], ny = y+dy[i];
      if (nx>=1 && nx<=n && ny>=1 && ny<=m && s[nx][ny]!=s[x][y] && dis[nx]
[ny]==-1) {
        q.push({nx, ny}); dis[nx][ny] = dis[x][y] + 1;
      }
    }
  }

  cout << dis[n][m] << endl;
  return 0;
}
```

马的遍历

马的遍历方式, 需要遍历周围的 8 个方向即可

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 400 + 5;
int w[maxn][maxn];
struct node {
  int x, y;
};
int dx[] = {-2, -2, -1, -1, 1, 1, 2, 2};
int dy[] = {-1, 1, -2, 2, -2, 2, -1, 1};

int main()
{
  int n, m, sx, sy; cin >> n >> m >> sx >> sy;

  memset(w, -1, sizeof(w));
  queue<node> q; q.push({sx, sy}); w[sx][sy] = 0;
  while (!q.empty()) {
```

```
    node u = q.front(); q.pop();
    int x = u.x, y = u.y;
    for (int i = 0; i < 8; ++i) {
      int nx = x+dx[i], ny = y+dy[i];
      if (nx>=1 && nx<=n && ny>=1 && ny<=m && w[nx][ny]==-1) {
        q.push({nx, ny}); w[nx][ny] = w[x][y] + 1;
      }
    }
  }

  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) printf("%d ", w[i][j]);
    cout << endl;
  }
  return 0;
}
```

[USACO07FEB] Bronze Lilypad Pond B

遍历方向也是 8 个方向, 这 8 个方向应该是 dx[] = {-m1, -m1, -m2, -m2, m2, m2, m1, m1}, dy[] = {-m2, m2, -m1, m1, -m1, m1, -m2, m2}

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 1000 + 5;
int a[maxn][maxn], w[maxn][maxn];
struct node {
  int x, y;
};

int main()
{
  int n, m, m1, m2; cin >> n >> m >> m1 >> m2;
  int dx[] = {-m1, -m1, -m2, -m2, m2, m2, m1, m1};
  int dy[] = {-m2, m2, -m1, m1, -m1, m1, -m2, m2};
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) cin >> a[i][j];
  }

  memset(w, -1, sizeof(w));
  queue<node> q;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) {
      if (a[i][j] == 3) q.push({i, j}), w[i][j] = 0;
    }
  }

  while (!q.empty()) {
    node u = q.front(); q.pop();
```

```
    int x = u.x, y = u.y;
    for (int i = 0; i < 8; ++i) {
      int nx = x+dx[i], ny = y+dy[i];
      if (nx>=1 && nx<=n && ny>=1 && ny<=m && a[nx][ny]!=0 && a[nx][ny]!=2 &&
w[nx][ny]==-1) {
        q.push({nx, ny}); w[nx][ny] = w[x][y] + 1;
      }
    }
  }

  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) {
      if (a[i][j] == 4) cout << w[i][j] << endl;
    }
  }
  return 0;
}
```

跳跃机器人

点 i 可以花一步走到点 i+1, i-1, i*2 这 3 个点

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e6 + 5;
int dis[maxn];

int main()
{
  int n; cin >> n;
  memset(dis, -1, sizeof(dis));
  queue<int> q; q.push(1); dis[1] = 0;
  while (!q.empty()) {
    int u = q.front(); q.pop();
    if (u == n) break;
    if (u-1>=1 && dis[u-1]==-1) { dis[u-1] = dis[u]+1; q.push(u-1); }
    if (u+1<=n && dis[u+1]==-1) { dis[u+1] = dis[u]+1; q.push(u+1); }
    if (u*2<=n && dis[u*2]==-1) { dis[u*2] = dis[u]+1; q.push(u*2); }
  }
  cout << dis[n] << endl;
  return 0;
}
```