

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1
по «Алгоритмам и структурам данных»

Выполнил:

Студент группы Р3210

Кадыров А.Х.

Преподаватели:

Косяков М.С.

Тараканов Д.С.

Санкт-Петербург

2019

Задача №1401 «Игроки»

Пояснение к примененному алгоритму:

Мы делим каждый раз нашу задачу на 4, делим поле на 4 части, пока размер одной такой части не будет 2 x 2. Сначала доходим до части, где стоит ноль, и зарисовываем остальные три точки. После (когда возвращается из квадрата 2 x 2 в исходный квадрат 4 x 4), зарисовываем три точки, в зависимости от того, где находился наш квадрат 2 x 2. Далее зарисовываем все остальные.

Код:

```
#include <iostream>
```

```
using namespace std;
```

```
int x0, y0, paint;
```

```
int table[520][520];
```

```
int n;
```

```
void TypeOfFhape(int x, int y, int type) {
```

```
    if (type == 1) {
```

```
        paint++;
```

```
        table[x][y] = table[x][y + 1] = table[x + 1][y] = paint;
```

```
    } else if (type == 2) {
```

```
        paint++;
```

```
        table[x][y] = table[x][y + 1] = table[x + 1][y + 1] = paint;
```

```
    } else if (type == 3) {
```

```
        paint++;
```

```
        table[x][y + 1] = table[x + 1][y] = table[x + 1][y + 1] = paint;
```

```
    } else if (type == 4) {
```

```
        paint++;
```

```
        table[x][y] = table[x + 1][y] = table[x + 1][y + 1] = paint;
```

```
    }
```

```
}
```

```
void draw(int x, int y, int det, int type) {
```

```

if (det == 1) {
    if ((x0 == x) && (y0 == y)) {
        TypeOfHape(x, y, 3);
    } else if ((x0 == x) && (y0 == y + 1)) {
        TypeOfHape(x, y, 4);
    } else if ((x0 == x + 1) && (y0 == y)) {
        TypeOfHape(x, y, 2);
    } else if ((x0 == x + 1) && (y0 == y + 1)) {
        paint++;
        table[x][y] = table[x][y + 1] = table[x + 1][y] = paint;
    } else if (type == 1) {
        TypeOfHape(x, y, 1);
    } else if (type == 2) {
        TypeOfHape(x, y, 2);
    } else if (type == 3) {
        TypeOfHape(x, y, 3);
    } else if (type == 4) {
        TypeOfHape(x, y, 4);
    }
} else {
    int d2 = det / 2;
    if (type == 0) {
        /// 1
        if ((x <= x0) && (x0 <= x + d2) && (y <= y0) && (y0 <= y + d2)) {
            draw(x, y, d2, 0);
            draw(x, y + d2 + 1, d2, 2);
            draw(x + d2 + 1, y + d2 + 1, d2, 3);
            draw(x + d2 + 1, y, d2, 4);
            paint++;
            table[x + d2 + 1][y + d2] = paint;
            table[x + d2 + 1][y + d2 + 1] = paint;
            table[x + d2][y + d2 + 1] = paint;
        }
    }
}

```

/// 2

```
else if ((x <= x0) && (x0 <= x + d2) && (y + d2 + 1 <= y0) &&
        (y0 <= y + det)) {
    draw(x, y, d2, 1);
    draw(x, y + d2 + 1, d2, 0);
    draw(x + d2 + 1, y + d2 + 1, d2, 3);
    draw(x + d2 + 1, y, d2, 4);
    paint++;
    table[x + d2][y + d2] = paint;
    table[x + d2 + 1][y + d2] = paint;
    table[x + d2 + 1][y + d2 + 1] = paint;
}
```

/// 3

```
else if ((x + d2 + 1 <= x0) && (x0 <= x + det) && (y + d2 + 1 <= y0) &&
        (y0 <= y + det)) {
    draw(x, y, d2, 1);
    draw(x, y + d2 + 1, d2, 2);
    draw(x + d2 + 1, y + d2 + 1, d2, 0);
    draw(x + d2 + 1, y, d2, 4);
    paint++;
    table[x + d2][y + d2] = paint;
    table[x + d2 + 1][y + d2] = paint;
    table[x + d2][y + d2 + 1] = paint;
}
```

/// 4

```
else if ((x + d2 + 1 <= x0) && (x0 <= x + det) && (y <= y0) &&
        (y0 <= y + d2)) {
    draw(x, y, d2, 1);
    draw(x, y + d2 + 1, d2, 2);
    draw(x + d2 + 1, y + d2 + 1, d2, 3);
    draw(x + d2 + 1, y, d2, 0);
    paint++;
    table[x + d2][y + d2] = paint;
```

```

    table[x + d2][y + d2 + 1] = paint;
    table[x + d2 + 1][y + d2 + 1] = paint;
}
} else if (type == 1) {
    /// term 3
    draw(x, y, d2, 1);
    draw(x, y + d2 + 1, d2, 2);
    draw(x + d2 + 1, y + d2 + 1, d2, 1);
    draw(x + d2 + 1, y, d2, 4);
    paint++;
    table[x + d2][y + d2] = paint;
    table[x + d2 + 1][y + d2] = paint;
    table[x + d2][y + d2 + 1] = paint;
} else if (type == 2) {
    /// term 4
    draw(x, y, d2, 1);
    draw(x, y + d2 + 1, d2, 2);
    draw(x + d2 + 1, y + d2 + 1, d2, 3);
    draw(x + d2 + 1, y, d2, 2);
    paint++;
    table[x + d2][y + d2] = paint;
    table[x + d2][y + d2 + 1] = paint;
    table[x + d2 + 1][y + d2 + 1] = paint;
} else if (type == 3) {
    /// term 1
    draw(x, y, d2, 3);
    draw(x, y + d2 + 1, d2, 2);
    draw(x + d2 + 1, y + d2 + 1, d2, 3);
    draw(x + d2 + 1, y, d2, 4);
    paint++;
    table[x + d2 + 1][y + d2] = paint;
    table[x + d2 + 1][y + d2 + 1] = paint;
    table[x + d2][y + d2 + 1] = paint;
}

```

```

    } else if (type == 4) {
        /// term 2
        draw(x, y, d2, 1);
        draw(x, y + d2 + 1, d2, 4);
        draw(x + d2 + 1, y + d2 + 1, d2, 3);
        draw(x + d2 + 1, y, d2, 4);
        paint++;
        table[x + d2][y + d2] = paint;
        table[x + d2 + 1][y + d2] = paint;
        table[x + d2 + 1][y + d2 + 1] = paint;
    }
}
}

```

```

int main() {
    cin >> n >> x0 >> y0;
    paint = 0;
    n = (1 << n);
    draw(1, 1, n - 1, 0);
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            cout << table[i][j] << " ";
        }
        cout << '\n';
    }
    return 0;
}

```