

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №5
По дисциплине « **Алгоритмы и структуры данных** »
Тема: «**Поиск с возвратом (Backtracking)**»

Выполнил:
Студент 2 курса
Группы ПО-11(2)
Сымоник И.А
Проверила:
Глущенко Т.А

Цель работы: изучить алгоритм поиска с возвратом.

Ход работы

Задание 1. Написать программу решения задачи о количестве островов: **200. Number of Islands** с ресурса LeetCode

Исходный код:

```
class Solution {
public:

    void backTrack(std::vector < std::vector<char>>& field,
                  const int x, const int y)
    {
        if (field[x][y] == '1')
        {
            field[x][y] = '2';
        }
        else
            return;

        if (x + 1 < field.size()) {
            backTrack(field, x + 1, y);
        }
        if (y + 1 < field[0].size()) {
            backTrack(field, x, y + 1);
        }
        if (x - 1 >= 0) {
            backTrack(field, x - 1, y);
        }
        if (y - 1 >= 0) {
            backTrack(field, x, y - 1);
        }
    }

    int numIslands(vector<vector<char>>& grid) {

        int countOfIslands = 0;

        for (int i = 0; i < grid.size(); i++)
        {
            for (int j = 0; j < grid[0].size(); j++)
```

```

    {
        if (grid[i][j] == '1')
        {
            backTrack(grid, i, j);
            countOfIslands++;
        }
    }
}
return countOfIslands;
}
};

```

Результат:

Runtime

20ms

Beats 97.77% of users with C++

Memory

12.62MB

Beats 73.64% of users with C++

Задание 2. Описать алгоритм решения задачи о 8 ферзях.

1. Создать пустую шахматную доску размером 8x8.
2. Начать с первого столбца (от 0 до 7) и разместить первого ферзя на первой строке (строке 0).
3. Поместить следующих ферзей, начиная со второго столбца (от 1 до 7), и двигаться слева направо по каждой строке (от 0 до 7).
4. В каждой позиции проверить, не находится ли ферзь под атакой другого ферзя.
5. Проверить, что нет другого ферзя в той же строке.
6. Проверить, что нет другого ферзя на одной диагонали с текущей позицией
7. Если текущая позиция удовлетворяет условиям, поместить ферзя на эту позицию.
8. Если каждый ферзь успешно размещен на доске, вывести решение.
9. Если не удалось разместить ферзя в текущей позиции, перейти к предыдущей позиции и попытаться разместить его в новом месте.
10. Повторять шаги 4-7 до тех пор, пока не будут размещены все ферзи или пока не будут перебраны все возможные варианты размещения ферзей.

Контрольные вопросы:

1. Если бы данные задачи решались полным перебором (brute force), чему бы могла быть равна временная сложность такой реализации?

Если бы задачи решались полным перебором, то временная сложность равнялась $O(n^2)$.

2. Какой из поисков на графах применяется в данных задачах?

В данных задачах применяется поиск в глубину. Поиск в глубину (DFS) — алгоритм обхода графа, который движется от начальной вершины как можно дальше сначала по первому ребру, потом по второму, и так далее.

3. Чему равна временная сложность поиска в глубину?

Временная сложность поиска в глубину равна — $O(V + E)$, где V — количество вершин, а E — количество ребер

4. Чему равна временная сложность задачи о количестве островов?

Временная сложность задачи о количестве островов равна — $O(m \cdot n)$, где m — количество строк, а n — количество столбцов матрицы.

5. Если в задаче о лабиринте есть несколько выходов из него, чем тогда определяется выбор варианта выхода?

Выбор варианта выхода зависит от реализации алгоритма. Выбор может осуществляться на основании следующих факторов:

1. Близость: Если существует ближайший выход от текущей позиции, то стоит выбрать его.
2. Оптимальность: Вариант выхода выбирается на основе оценок или функции стоимости для каждого выхода.
3. Случайный выбор: Если все выходы равноценны, то можно использовать случайный выбор варианта выхода.

6. Какова альтернативная формулировка задачи о 8 ферзях и сколько решений она имеет?

Заполнить матрицу размером 8x8 нулями и единицами таким образом, чтобы сумма всех элементов матрицы была равна 8, при этом сумма элементов ни в одном столбце, строке или диагональном ряде не превышала единицы.

Задача имеет 92 решения.

Вопросы по задаче 200:

1. Чему равна временная сложность алгоритма?

Временная сложность алгоритма – $O(n * m)$, где n – количество столбцов, m – количество строк.

2. Какой алгоритм на графах мы использовали в этой задаче?

Алгоритм поиска в глубину.

3. Указать порядок обработки ячеек для острова ниже.

1	2	5	
	3	4	
6			
7	8		

Вывод: изучили алгоритм поиска с возвратом.