

Министерство образования и науки Российской Федерации федеральное государственное
автономное образовательное учреждение высшего образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет «Программной инженерии и компьютерной техники.»

Алгоритмы и структуры данных

Лабораторная работа №1
Timus

Выполнил

Григорьев Давид Владимирович

Группа: Р3215

Преподаватели

Косяков Михаил Сергеевич

Тараканов Денис Сергеевич

Содержание

1	1005	1
1.1	Пояснение к примененному алгоритму	1
1.2	Код алгоритма	1
2	1401	2
2.1	Пояснение к примененному алгоритму	2
2.2	Код алгоритма	2

1 1005

1.1 Пояснение к примененному алгоритму

Time complexity: $O(2^n)$

1.2 Код алгоритма

```
#include <iostream>
#include <vector>

int main() {
    int number_of_stones = 0;
    std::cin >> number_of_stones;
    std::vector<int> stones(number_of_stones);
    int total_sum = 0;
    for (int i = 0; i < number_of_stones; ++i) {
        std::cin >> stones[i];
        total_sum += stones[i];
    }

    int max_sum = 0;
    int target = total_sum / 2;

    for (int mask = 0; mask < (1 << number_of_stones); mask++) {
        int current = 0;
        for (int i = 0; i < number_of_stones; ++i) {
            if ((mask & (1 << i)) != 0) {
                current += stones[i];
                if (current > target) {
                    break;
                }
            }
        }
        if (current > max_sum && current <= target) {
            max_sum = current;
        }
    }

    std::cout << total_sum - (2 * max_sum) << '\n';

    return 0;
}
```

2 1401

2.1 Пояснение к примененному алгоритму

Time complexity: $O(n^4)$

2.2 Код алгоритма

```
/* Problem 1401: Tiling with L-Shaped Trominoes  
* Efficiency: Operates in  $O(4^n)$  time, efficiently covering all cells except the initial hole  
*/  
  
#include <array>  
#include <iostream>  
const int GridSize = 512;  
  
void placeTiles(  
    int size,  
    int x,  
    int y,  
    int hole_x,  
    int hole_y,  
    int& tile_counter,  
    std::array<std::array<int, GridSize>, GridSize>& grid  
) {  
    if (size == 2) {  
        // Fill 2x2 grid around the hole  
        for (int dx = 0; dx < 2; ++dx) {  
            for (int dy = 0; dy < 2; ++dy) {  
                if (x + dx != hole_x || y + dy != hole_y) {  
                    grid[x + dx][y + dy] = tile_counter++ / 3;  
                }  
            }  
        }  
        return;  
    }  
  
    int half = size / 2;  
    // Place central trominoes in three quadrants without the hole  
    for (int i = 0; i < 2; ++i) {  
        for (int j = 0; j < 2; ++j) {  
            int quad_x = x + (i * half);  
            int quad_y = y + (j * half);  
            if (quad_x > hole_x || hole_x >= quad_x + half || quad_y > hole_y ||  
                hole_y >= quad_y + half) {  
                grid[x + half - 1 + i][y + half - 1 + j] = tile_counter++ / 3;  
            }  
        }  
    }  
  
    // Process each quadrant  
    for (int i = 0; i < 2; ++i) {  
        for (int j = 0; j < 2; ++j) {  
            int quad_x = x + (i * half);  
            int quad_y = y + (j * half);
```

```

        if (quad_x <= hole_x && hole_x < quad_x + half && quad_y <= hole_y &&
            hole_y < quad_y + half) {
            placeTiles(half, quad_x, quad_y, hole_x, hole_y, tile_counter, grid);
        } else {
            int new_hole_x = x + half - 1 + i;
            int new_hole_y = y + half - 1 + j;
            placeTiles(half, quad_x, quad_y, new_hole_x, new_hole_y, tile_counter, grid);
        }
    }
}

int main() {
    int tile_counter = 3;
    std::array<std::array<int, GridSize>, GridSize> grid{};

    int n = 0;
    int hole_row = 0;
    int hole_col = 0;

    std::cin >> n;
    std::cin >> hole_row;
    std::cin >> hole_col;

    int grid_size = 1 << n; // Compute 2^n
    placeTiles(grid_size, 0, 0, hole_row - 1, hole_col - 1, tile_counter, grid);

    for (int i = 0; i < grid_size; ++i) {
        for (int j = 0; j < grid_size; ++j) {
            std::cout << grid[i][j] << " ";
        }
        std::cout << '\n';
    }
    return 0;
}

```