

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Решения задач тимус по  
Алгоритмам и структурам данных

Работу выполнил:

Гаврилин О.С.

Группа:

P3230

Санкт-Петербург,

2025

## 1. 1494

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

using namespace std;
void solve() {
    int totalBalls;
    cin >> totalBalls;

    vector<int> sequence(totalBalls);
    for (int &ball : sequence) {
        cin >> ball;
    }

    stack<int> pocket;
    int nextBall = 1, index = 0;

    for (; nextBall <= totalBalls; ++nextBall) {
        pocket.emplace(nextBall);

        while (!pocket.empty() && pocket.top() ==
sequence[index]) {
            pocket.pop();
            ++index;
        }
    }

    cout << (index == totalBalls ? "Not a proof" :
"Cheater") << '\n';
}

int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(nullptr);
    solve();
}
```

**Описание:** Каждый шар от 1 до N кладем в стек, пока не совпадет с текущим нужным шаром в sequence, если верх совпал, то убираем его. Если все шары достали верно, индекс дошел до общего количества то выводим Not a proof, иначе Cheater

**Сложность по времени:**  $O(n)$

## Сложность по памяти: $O(n)$

### 2. 1067

```
#include <iostream>
#include <map>
#include <string>
#include <sstream>

using namespace std;

void solve() {
    struct Folder {
        map<string, Folder*> inside;
    };

    Folder mem[50001];
    int next = 1;

    auto add = [&](Folder* root, const string& path) {
        string name;
        stringstream ss(path);
        while (getline(ss, name, '\\')) {
            if (!root->inside[name])
                root->inside[name] = &mem[next++];
            root = root->inside[name];
        }
    };

    auto show = [&](auto&& show, Folder* folder, int level)
-> void {
        for (auto& [name, sub] : folder->inside) {
            cout << string(level, ' ') << name << '\n';
            show(show, sub, level + 1);
        }
    };

    int count;
    cin >> count;
    Folder* root = &mem[0];

    while (count--) {
        string input;
        cin >> input;
        add(root, input);
    }
}
```

```

    }

    show(show, root, 0);
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    solve();
}

```

**Описание:** Каждая папка представляет собой узел, содержащий `map<string, Folder*> inside`, использую для поддержания сортировки по имени при обходе. Строка разбивается по символу '\\', проверяем в каждой части пути есть ли уже такая папка в `inside`, если нет, то создаем новую используя заготовленный массив. При выводе дерева используем рекурсивный обход, идем по всем дочерним папкам и печатаем имя с отступом.

**Сложность по времени:**  $O(T \log M)$

**Сложность по памяти:**  $O(n)$ ,  $n$  – количество уникальных папок в дереве.