Практикум 10 - Дървета

[Произволни, наредени, самобалансиращи се]

Задача 1. Напишете функция, която приема един аргумент – корен на дърво, чиито възли могат да имат произволен брой деца. Всеки възел пази стойност – цяло число.

Функцията трябва да намери онзи път от корена на дървото, до някое листо, който има най-малка сума на елементите в него. Този път трябва да се върне, като вектор от възлите в него. Ако има няколко такива пътя, може да се върне който и да е от тях. Напишете кратка програма, която демонстрира работата на функцията върху примерно дърво. По-долу са дадени представянето на дървото и прототипът на функцията, която трябва да реализирате. Те трябва да са точно такива, както са показани тук.

```
struct Node {
    int value;
    std::vector<Node> children;
};

std::vector<Node*> findCheapestPath(Node *root);
```

Задача 2. Напишете програма, която по подадено естествено число N и толкова на брой числа извежда на екрана числата в сортиран вид. Нека функцията в най-лошия случай да работи за време O(N*log(N)) и да използва O(1) допълнителна памет.
Забележка: Без да ползвате сортиращи алгоритми.

Задача 3. Намерете броя на поддърветата на двоично наредено дърво, на които всички възли са от даден интервал `[p, q]`.

Задача 4. Проверете дали стойностите на всички N възли на двоично наредено дърво са от даден интервал `[p, q]`.
Забележка: Нека алгоритъмът да работи за време O(log(N)) в най-лошия случай (ако дървото е “сравнително” балансирано).