

Алгоритмы и структуры данных

Домашняя работа

Неделя 3

Иван Алексеев М3139

11.10.2020

Задача 1

Решим задачу через дерево сравнений. У последовательности из n элементов будет всего $n!$ листьев (число перестановок). Тогда высота такого бинарного дерева с округлением вверх $\log_2(n!) \approx 7$.

Задача 2

Приведем один из примеров построения такой перестановки:

$$2, 3, 4, 5 \dots n, 1$$

В этом случае при любом обмене никакие два элемента не встанут на нужные места.

Всего существует $(n - 1)!$ таких перестановок. Почему это так: пусть мы рассматриваем какую-то перестановку с отсортированным префиксом длины i . Тогда для $(i + 1)$ -ого места у нас существует

$n - i - 1$ способа выбрать не минимальный элемент. Тогда всего таких перестановок $(n - 1)!$

Задача 3

Такая перестановка:

$$n, (n - 1), (n - 2), \dots, 2, 1$$

В этой перестановке максимальный элемент находится всегда слева с краю, поэтому он будет делать обмен с любым элементом справа не входящим в отсортированный суффикс.

Всего существует $(n - 1)!$ таких перестановок. Почему это так: пусть мы рассматриваем какую-то перестановку с отсортированным суффиксом длины i . Тогда для $(n - i)$ -ого места у нас существует $(n - i - 1)$ способ выбрать не максимальный элемент. Тогда всего таких перестановок $(n - 1)!$

Задача 4

а) Из перестановки для сортировки пузырьком в перестановку для сортировки выбором.

Назовем перестановку пузырьком x , а перестановку выбором - y . Тогда будем брать попарные элементы x_i и x_{i+1} . Пусть $j = x_i$. В j -ый элемент y кладем x_{i+1} . Получаем однозначное преобразование элемента из x_i в y_j . Следовательно алгоритм работает за линейное время.

б) Из перестановки для сортировки выбором в перестановку для сортировки пузырьком.

Назовем перестановку пузырьком y , а перестановку выбором - x . Для обратного алгоритма нужно чтобы 1 стояла на последнем месте. Если в x на i -ом месте стоит x_i , то в y элементы равные i и a_i соответственно стоят друг за другом. Тогда смотрим на первый элемент в x : элементы 1 и x_1 должны стоять друг за другом, а т.к. 1 стоит в конце, то x_1 - стоит первым в массиве y . Обращаемся к элементу с индексом x_1 , ставим его на место y_2 и т.д. Алгоритм работает за линейное время, т.к. на каждой итерации один обмен.