Теоретические задачи для недели №3

Александр Курилкин, ШАД HELPER

28 июля 2020 г.

Во всех задачах время работы может быть амортизированным или ожидаемым (то есть матожидание времени работы), если не указано иное.

№1

(a)

Сортировка называется стабильной, если она не меняет порядок элементов, которые равны (это имеет смысл, если сами объекты отличаются, то при этом функция их сравнения говорит, они равны). Какие из следующий сортировок являются стабильными: выбором, вставками, пузырьком, слиянием, быстрая, кучей (из задачи 2)?

(b)

Как любую сортировку сделать стабильной, потратив $O(n)$ дополнительной памяти?

№2

Докажите, что сортировка вставками работает за $O(n + inv)$, где inv — количество инверсий в массиве. Напомним, что инверсией называется такая пара i, j, что $i < j$ и $a_i > a_j$

№3

Давайте уметь поддерживать еще две операции с кучей: $get\_kth()$ — узнать k-й по величине элемент в куче, и $delete\_kth()$ — удалить k-й по величине элемент, при этом k одно и то же для всех запросов. Как реализовать их так, чтобы они работали за $O(\log n)$?

№4*

Изобретите сортировку на основе кучи, которая бы использовала $O(1)$ дополнительной памяти и гарантированно работала за $O(n \log n)$.

№5**

Докажите формально время работы $O(n \log n)$ для быстрой сортировки, то есть посчитайте матожидание времени ее работы по определению матожидания.

Подсказка: это делается по индукции, приняв время работы на n элементах за $T(n) = cn \log_2 n$, где c — какая-то **фиксированная** константа.

№6**

Давайте уметь поддерживать еще две операции с кучами: $init(x)$ — создать кучу из одного элемента x и $merge(heap_1, heap_2)$ — слить две кучу в одну корректную (так, чтобы в ней содержались все элементы из $heap_1$ и $heap_2$). Придумайте, как реализовать $merge$ так, чтобы он работал за $O(\log^2 n)$, где n — суммарное количество элементов во всех кучах.

Подсказка: посмотрите в самое-самое начало листка.