Тимсорт

Санду Р.А.

15 декабря 2017 г.

Не сложно заметить, что первый этап алгоритма линеен: в худшем случае данные перемешаны настолько, что каждый ран (от англ. run) придётся добивать до минимальной длинны сортировкой вставками. Но тогда это займёт $O(\frac{n}{min\_run} min\_run^2) = O(n \cdot min\_run)$ времени, где min_run — константа порядка 2^5

При добавлении рана R в стек будем класть на него по $\operatorname{coins} R = \operatorname{hgt} R \cdot O(\operatorname{len} R)$ монет, где $\operatorname{hgt} R$ — индекс элемента в стеке, где $\operatorname{hgt} R = 0$ означает, что ран в самом низу стека, а $\operatorname{len} R$ — длинна рана. Пусть после добавления нового рана нарушился один из инвариантов. Обозначим за S_i текущие раны в стеке, где S_0 — вершина стека, т.е. только что добавленный ран. Тогда инварианты имеют вид

$$\operatorname{len} S_1 > \operatorname{len} S_0, \tag{1}$$

$$\operatorname{len} S_2 \geq \operatorname{len} S_1 + \operatorname{len} S_0. \tag{2}$$

При слиянии будем класть монеты из сливаемых ранов на получившийся ран. Также будем поддерживать всё время работы инвариант

$$\operatorname{coins} S_i = \operatorname{hgt} R \cdot O(\operatorname{len} S_i). \tag{3}$$

Первый случай: нарушился инвариант (1), $S_1 \leq S_0$. Тогда сливатся будут S_1 и минимальный по длинне из S_0 и S_2, а значит нужно $\operatorname{len} S_1 + \min(\operatorname{len} S_0, \operatorname{len} S_2)$ монет на операцию слияния. Но так как в данном случае $\operatorname{len} S_1 < \operatorname{len} S_0$ и $\min(\operatorname{len} S_0, \operatorname{len} S_2) \leq \operatorname{len} S_0$, имеем

$$\operatorname{len} S_1 + \min(\operatorname{len} S_0, \operatorname{len} S_2) = O(\operatorname{len} S_0). \tag{4}$$

Заметим также, что при любом из слияний $\operatorname{hgt} S_0$ гарантированно уменьшится на единицу, а значит мы сможем использовать $O(\operatorname{len} S_0)$ монет, не нарушая инвариант (3), и тогда в данном случае операция оплачена.

Второй случай: нарушился инвариант (2), $\operatorname{len} S_2 < \operatorname{len} S_1 + \operatorname{len} S_0$. Разберём первый подслучай: $\operatorname{len} S_2 < \operatorname{len} S_0$, сливать будем S_2 и S_1. Тогда получаем из нарушения $\operatorname{len} S_2 = O(\operatorname{len} S_1 + \operatorname{len} S_0)$, и прибавив к обоим частям $\operatorname{len} S_1$, получаем

$$\operatorname{len} S_1 + \operatorname{len} S_0 = O(\operatorname{len} S_0 + \operatorname{len} S_1). \tag{5}$$

А так как hgt S_0 и hgt S_1 уменьшатся, получается, что у нас есть те самые $O(\operatorname{len} S_0)+O(\operatorname{len} S_1)$ монет, чтобы оплатить операцию.

Разберём второй подслучай: $\operatorname{len} S_2 > \operatorname{len} S_0$, сливаем S_1 и S_0. Рассмотрим, что произойдёт после слияния, обозначив за $S_i \cup S_j$ слитые раны S_i и S_j. Очевидно, что $\operatorname{len}(S_0 \cup S_1) > \operatorname{len} S_2$, а значит следующее слияние обязательно будет. Скажем, что в этом случае проверки инвариантов мы проводить не будем, а просто сразу начнём востанавливать его дальше. Проходить оно будет по первому случаю, и нам придётся на него потратить $O(\operatorname{len}(S_0 \cup S_1)) = O(\operatorname{len} S_0 + \operatorname{len} S_1)$ монет. Но тогда за счёт увеличения константы мы можем оплатить этими монетами и слияние S_0 и S_1.

Таким образом получается, что все операции оплачены. Учётная стоимость всего второго этапа получается

$$\sum_{i=0}^{r} l_i h_i \leq \sum_{i=0}^{r} l_i \bar{h} \leq n\bar{h}, \tag{6}$$

где l_i — длинна рана, добавленного i-м, h_i — высота стека в момент добавления, n — кол-во элементов в массиве, а $\bar{h}$ — верхняя оценка на высоту стека.

Докажем, что $\bar{h} = O(\log n)$. Из инварианта (2) получается, что в каждый момент времени $\operatorname{len} S_i$ ограничено снизу F_i — i-м числом фибоначчи, где индексация S_i идёт с низа стека. Но частичные суммы $\operatorname{len} S_i$ ограниченны F_{i+2}, а по формуле Бине $F_{i+2} = O(exp(i+2))$. Частичные суммы никогда не превысят n, а значит и нижняя грань тоже будет меньше или равна n. Тогда получаем, что $exp(\bar{h}) = O(n)$, а значит $\bar{h} = O(\log n)$, что и требовалось доказать.

Далее, о том, в каком порядке востанавливается инвариант. Первичных проверок после добавления новго рана не больше, чем самих ранов, а значит оплатить их можно просто положив на константу монеток на каждый ран больше. Если инвариант сломался, то после его исправления достаточно посмотреть на 5 верхних ранов в стеке чтобы понять, сломан ли инвариант. Все эти проверки можно оплатить, положив на каждый ран на константу монеток больше, ведь после каждого исправления у нас есть $O(\operatorname{len} S_i)$ монет на то, чтобы оплатить востановление и проверку. Тогда каждую проверку можно оплатить константой монет на каждом из этих ранов. Таким образом, все проверки выполнения инварианта тоже оплачены.

Значит второй этап работает за $O(n \log n)$. Из ограничение на кол-во слияний так же следует, что и третий этап работает за $O(n \log n)$, и получаем, что суммарна весь алгоритм работает за $O(n \log n)$.