«Разделяй и властвуй»: сортировка слиянием

Александр Куликов

Онлайн-курс «Алгоритмы: теория и практика. Методы»
http://stepic.org/217

Постановка задачи

Сортировка

Вход: массив $A[1 \dots n]$.

Выход: перестановка $A'[1 \dots n]$ элементов массива $A[1 \dots n]$, в которой элементы упорядочены по неубыванию: $A'[1] \leq A'[2] \leq \cdots \leq A'[n]$.

Замечания

- Алгоритм имеет доступ к оракулу сравнения. Считаем, что сравнение занимает константное время.
- Если $A = A'$, то алгоритм сортирует на месте.

Сортировка вставками

Процедура InsertionSort($A[1 \ldots n]$)

```
для i от 2 до n:
  j ← i
  пока j > 1 и A[j] < A[j − 1]:
    обменять A[j] и A[j − 1]
    j ← j − 1
```

Корректность и время работы

- Корректность следует из выполнения следующего инварианта: после i итераций внешнего цикла подмассив $A[1 \ldots i]$ упорядочен (по неубыванию).
- Время работы: $1 + 2 + \cdots + (n-1) = \Theta(n^2)$.
- Массив сортируется на месте.

Сумма арифметической прогрессии

Лемма

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$

Сумма арифметической прогрессии

Лемма

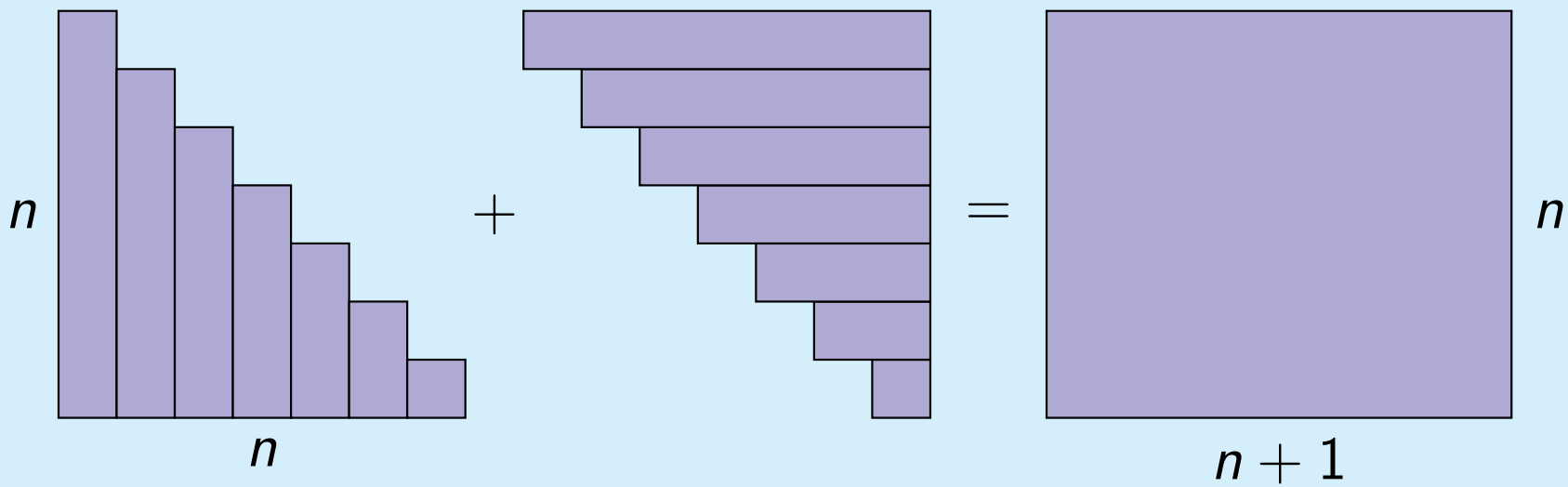
$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$

Доказательство

$$\begin{array}{ccccccc} 1 & 2 & 3 & \cdots & n-1 & n & \\ n & n-1 & n-2 & \cdots & 2 & 1 & \\ \hline n+1 & n+1 & n+1 & \cdots & n+1 & n+1 & = n(n+1) \end{array} \quad \square$$

Альтернативное доказательство

Доказательство



□

Сортировка слиянием

Процедура MergeSort(A, ℓ, r)

если $\ell < r$:

$m \leftarrow \lfloor \frac{\ell + r}{2} \rfloor$

Merge(MergeSort(A, ℓ, m), MergeSort$(A, m + 1, r)$)

Сортировка слиянием

Процедура MergeSort(A, ℓ, r)

если $\ell < r$:

$m \leftarrow \lfloor \frac{\ell + r}{2} \rfloor$

$\mathtt{Merge}(\mathtt{MergeSort}(A, \ell, m), \mathtt{MergeSort}(A, m + 1, r))$

Процедура Merge

- Сливает два упорядоченных массива в один.
- Работает за линейное время от суммы размеров данных двух массивов: первым элементом результирующего массива будет меньший из первых элементов данных массивов, оставшаяся часть может быть заполнена рекурсивно.

Время работы

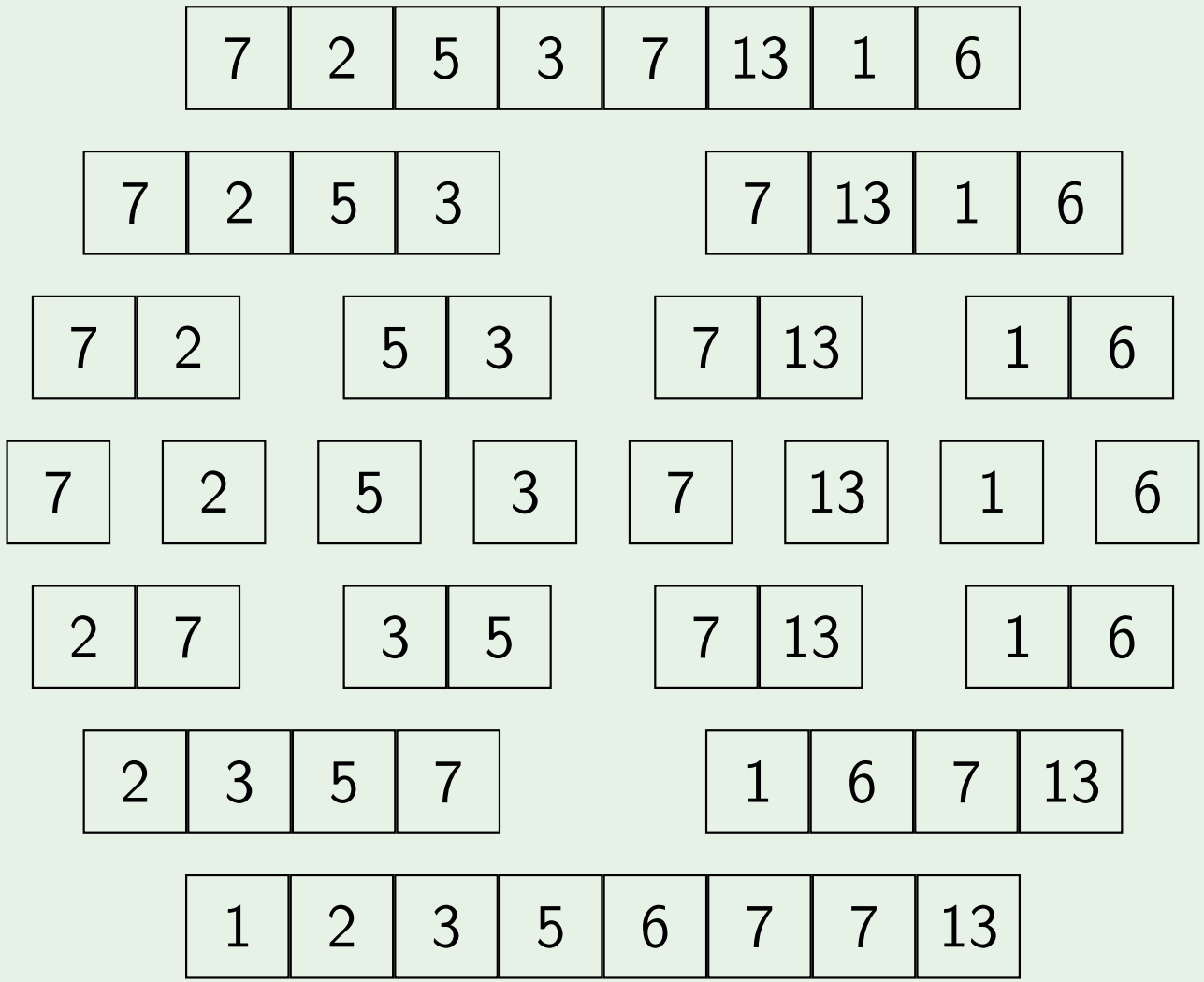
Лемма

Время работы алгоритма сортировки слиянием — $O(n \log n)$.

Доказательство

- Рекуррентное соотношение на время работы: $T(n) = 2T(n/2) + O(n)$.
- В дереве рекурсии будет $\log_2 n$ уровней. Суммарный размер задач на каждом уровне — n, работа на каждом уровне — $O(n)$.

Пример


7 2 5 3 7 13 1 6
7 2 5 3
7 13 1 6
7 2
5 3
7 13
1 6
7
2
5
3
7
13
1
6
2 7
3 5
7 13
1 6
2 3 5 7
1 6 7 13
1 2 3 5 6 7 7 13

Итеративная сортировка слиянием

Функция IterativeMergeSort($A[1 \ldots n]$)

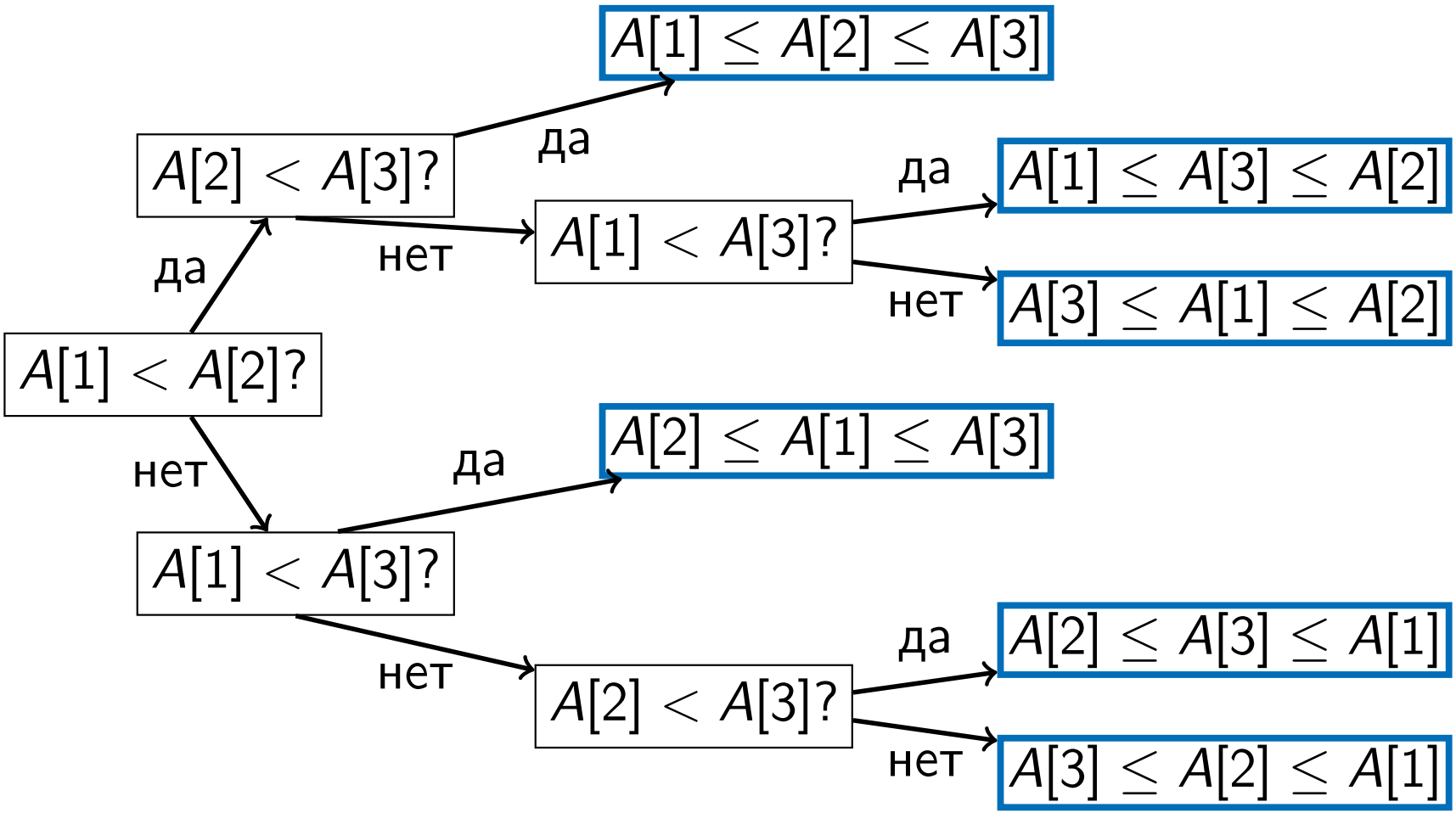
$Q \leftarrow [\,]$ {пустая очередь}
для i от 1 до n:
  PushBack($Q, [A[i]]$)
пока $|Q| > 1$:
  PushBack(Q, Merge(PopFront(Q), PopFront(Q)))
вернуть PopFront(Q)

Нижняя оценка для сортировки сравнениями

Лемма

Любой корректный алгоритм сортировки, основанный на сравнениях элементов, делает $\Omega(n \log n)$ сравнений в худшем случае на массиве размера n.

Дерево сравнений



Доказательство

- число листьев ℓ в дереве $\geq n!$ (число перестановок)

Доказательство

- число листьев ℓ в дереве $\geq n!$ (число перестановок)
- время работы алгоритма (число сделанных сравнений) в худшем случае не меньше глубины дерева d

Доказательство

- число листьев ℓ в дереве $\geq n!$ (число перестановок)
- время работы алгоритма (число сделанных сравнений) в худшем случае не меньше глубины дерева d
- $d \geq \log_2 \ell$ (или, что то же самое, $2^d \geq \ell$)

Доказательство

- число листьев ℓ в дереве $\geq n!$ (число перестановок)
- время работы алгоритма (число сделанных сравнений) в худшем случае не меньше глубины дерева d
- $d \geq \log_2 \ell$ (или, что то же самое, $2^d \geq \ell$)
- таким образом, время работы не меньше

$$\log_2(n!) = \Omega(n \log n)$$

Доказательство

- число листьев ℓ в дереве $\geq n!$ (число перестановок)
- время работы алгоритма (число сделанных сравнений) в худшем случае не меньше глубины дерева d
- $d \geq \log_2 \ell$ (или, что то же самое, $2^d \geq \ell$)
- таким образом, время работы не меньше

$$\log_2(n!) = \Omega(n \log n)$$

  - формула Стирлинга: $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$

Доказательство

- число листьев ℓ в дереве $\geq n!$ (число перестановок)
- время работы алгоритма (число сделанных сравнений) в худшем случае не меньше глубины дерева d
- $d \geq \log_2 \ell$ (или, что то же самое, $2^d \geq \ell$)
- таким образом, время работы не меньше

$$\log_2(n!) = \Omega(n \log n)$$

  - формула Стирлинга: $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$
  - $n! > \left(\frac{n}{2}\right)^{n/2}$, поэтому $\log(n!) = \Omega(n \log n)$ □

Заключение

- Наивный алгоритм сортировки имеет время работы $O(n^2)$.
- Алгоритм сортировки слиянием, основанный на методе «разделяй и властвуй», работает за время $O(n \log n)$. Использует $O(n)$ дополнительной памяти (для слияния массивов).
- Любой алгоритм, сортирующий сравнениями, в худшем случае делает $\Omega(n \log n)$ сравнений. Поэтому алгоритм сортировки слиянием асимптотически оптимален.