

Терпеливая (Пасьянсная) сортировка:

В ходе презентации: основные принципы работы алгоритма, его преимущества и недостатки, а также области, где он демонстрирует свою максимальную эффективность.



Принцип работы Терпеливой сортировки

Фаза 1: Построение стопок

Алгоритм начинается с создания стопок. Каждое число из входного массива рассматривается по очереди. Если текущее число меньше вершины какой-либо существующей стопки, оно помещается на эту стопку. В противном случае, создается новая стопка, и число становится ее первой картой.

Фаза 2: Слияние стопок

После того как все числа распределены по стопкам, начинается фаза слияния. На этом этапе мы многократно извлекаем наименьший элемент со всех вершин стопок и добавляем его в результирующий отсортированный массив. Это продолжается до тех пор, пока все стопки не опустеют.



Преимущества Терпеливой сортировки

Эффективность на почти отсортированных данных

Терпеливая сортировка демонстрирует впечатляющую производительность на данных, которые уже частично отсортированы в отличие от той же быстрой сортировки)

Связь с НВП

Количество стопок, образующихся на первом этапе, напрямую соответствует длине самой длинной возрастающей подпоследовательности во входном массиве.

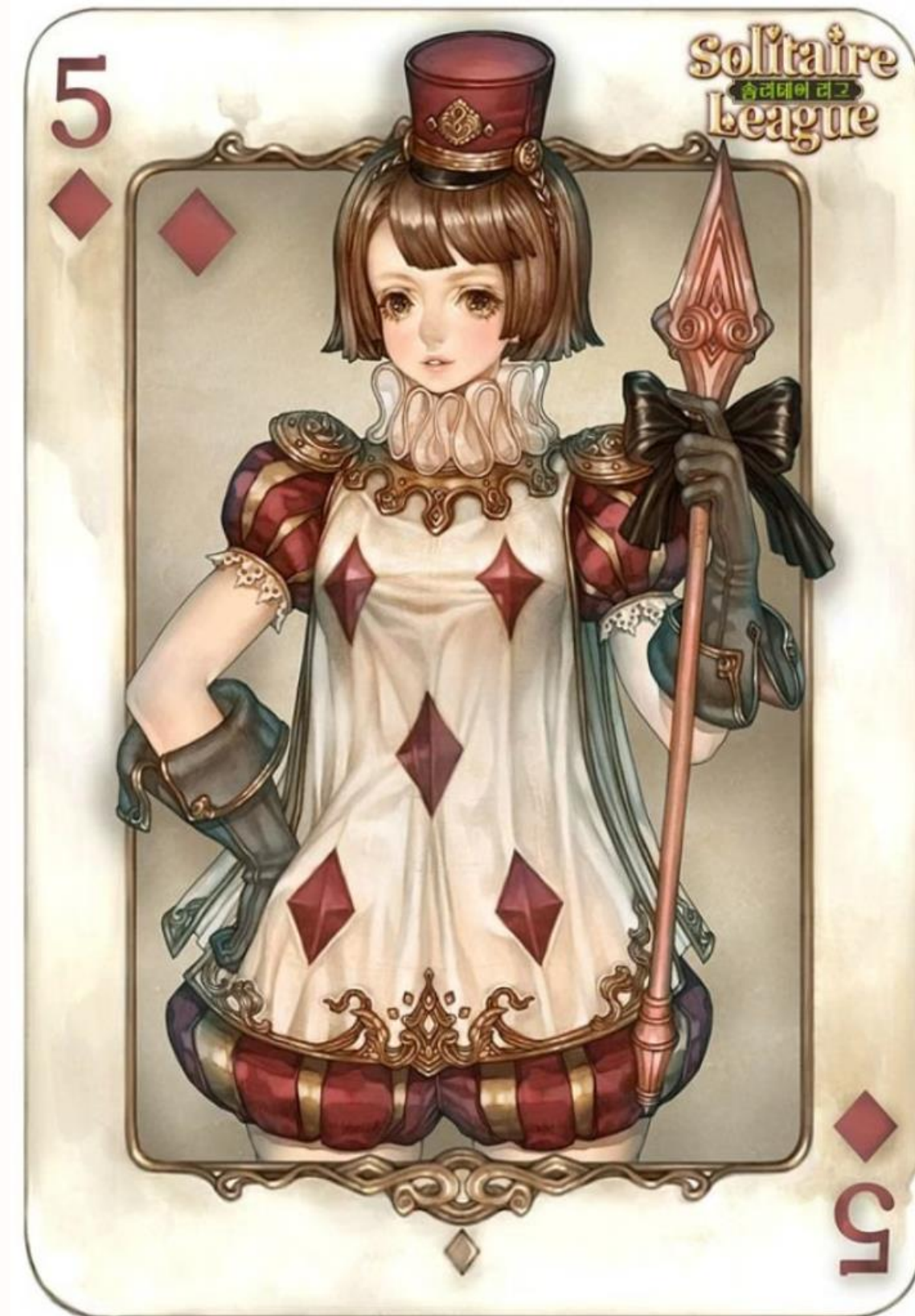
Недостатки Терпеливой сортировки

При большом кол-ве инверсии

Если входной массив содержит большое количество инверсий, будет создано много стопок. Это приведет к замедлению работы алгоритма (накладные расходы и т.п.).

Требования к памяти (в ходе написания алгоритма прям очень часто создается массив, хотя может это в частности и зависит от реализации)

В худшем случае, когда каждый элемент образует новую стопку, объем необходимой памяти может быть сравним с размером входного массива, что так-то недостаток



Сравнение с другими алгоритмами

Средняя временная сложность	$O(N \log N)$	$O(N \log N)$	$O(N \log N)$
Худшая временная сложность	$O(N^2)$ (в некоторых случаях)	$O(N^2)$	$O(N \log N)$
Пространственная сложность	$O(N)$	$O(\log N)$	$O(N)$
Стабильность	Нет (по умолчанию)	Нет	Да

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Bubble Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
Mergesort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
Quicksort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(\log(n))$
Heapsort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(1)$
Counting Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n+k)$	$O(k)$
Radix Sort	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$	$O(n+k)$
Bucket Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$	$O(n)$
Shell Sort	$\Omega(n \log(n))$	$\Theta(n(\log(n))^2)$	$O(n(\log(n))^2)$	$O(1)$

Применение Терпеливой сортировки

Поиск НВП

Упоминалось ранее (3 слайд)

Оптимизация баз данных

В некоторых сценариях, где данные поступают потоком и их необходимо эффективно сортировать для индексирования или запросов, терпеливая сортировка может быть использована для поддержания частично отсортированных структур данных.



Заключение и выводы

Терпеливая сортировка — это в какой-то степени уникальная, мемная и в каких-то случаях реально полезная сортировка)