

Лабораторная работа II. Сортировки.

Задача: написать улучшения к сортировкам с квадратичной асимптотической сложностью, провести декомпозицию на элементарные шаги и протестировать отдельные функции.

Шейкерная сортировка (3 - 5)

Напишите «шейкерную сортировку», которая является улучшением сортировки пузырьком путём дополнительного встречного прохода (от последнего элемента к первому). При написании программы проведите декомпозицию: напишите отдельные функции для прямого и обратного прохода и саму функцию сортировки, которая использует функции проходов. Напишите программы для тестирования трёх функций по отдельности.

```
1 void forward_step(  
2     unsigned arr[],  
3     unsigned const begin_idx, unsigned const end_idx);  
4 void backward_step(  
5     unsigned arr[],  
6     unsigned const begin_idx, unsigned const end_idx);  
7 void shaker_sort(  
8     unsigned arr[],  
9     unsigned const begin_idx, unsigned const end_idx);
```

Сортировка расчёской (6 - 7)

Напишите сортировку «расчёской» - улучшение для сортировки пузырьком с использованием предварительной сортировки с большим шагом. Немодифицированная сортировка пузырьком при проходе по массиву сравнивает правильность порядка соседних элементов, предполагается, что сортировка будет работать быстрее, если предварительно проверить порядок элементов с шагами $N/2$, $N/4$ и т.д. В конце проводится обычная сортировка пузырьком - проверка порядка с шагом 1 до тех пор пока массив не будет отсортирован. Напишите функцию проверки порядка элементов в массиве с заданным шагом. Используйте её в основной функции сортировки. Как изменяется время с увеличением числа элементов? Как изменяется количество перестановок с увеличением числа элементов? Постройте графики зависимости.

Сортировка Шелла (8 - 10)

Сортировкой Шелла называется усовершенствование сортировки вставками, путём предварительных грубых проходов (как в сортировке расчёской). В данном пункте вам предлагается реализовать сортировку Шелла и прямым измерением времени работы алгоритма определить асимптотическую сложность сортировки для различных вариантов последовательностей шагов:

1. $d_1 = N$, $d_{i+1} = d_i/2$ последний проход - все элементы ($d_{last} = 1$);
2. последовательности Хиббарда: $d_i = 2^i - 1$, находим наибольшее $i : 2^i - 1 \leq N$, с каждым шагом уменьшаем i ;
3. обратная последовательность чисел Фибоначчи: находим наибольшее число Фибоначчи $F_n : F_n \leq N$, с каждым шагом выбираем предыдущее число Фибоначчи.

Для каждой последовательности постройте график зависимости времени от количества элементов в массиве и определите показатель степени в асимптотике, для каждой последовательности подсчитайте среднее число потребовавшихся перестановок. Какая последовательность наиболее эффективна по времени или числу перестановок?