

Задачи для 1 курса (2021-2022 уч. год.)

3. Задачи на сортировку массивов

Требуется реализовать и протестировать различные алгоритмы сортировки. Тестирование означает следующее.

Каждый алгоритм сортировки надо реализовать в трех вариантах:

1. для массива значений типа `double` (или `int` для специальных сортировок)

```
void Sort1(double *a, int n);
```

2. для массива значений типа `double` с заданной функцией сравнения (соответственно, `int`)

```
void Sort2(double *a, int n, int (*cmp)(double, double));
```

3. для массива произвольных объектов с заданной функцией сравнения (по типу `qsort`)

```
void Sort3(void *a, int n, int elem_length, int (*cmp)(const void *, const void *));
```

Для каждой из этих реализаций надо измерить время сортировки (одного и того же) случайного массива длины N и вывести это время в виде сравнительной таблицы для все трех реализаций.

Тест должен также проверять фактическую упорядоченность отсортированного массива.

Как дополнительный бонус может быть предложено сравнить скорость работы вашей реализации с библиотечной функцией `qsort` (или алгоритмом `sort` для `vector`).

Как дополнительный бонус может быть предложено проверить скорость работы вашей реализации на специфических массивах (случайный, упорядоченный, частично упорядоченный, “волнообразный” и т.п.).

Время работы каждой сортировки замеряется с помощью функции `clock()` или любым другим подходящим способом.

Алгоритмы медленной сортировки:

1. Простая сортировка обмeнами (перестановка максимума).
2. Пузырьковая восходящая сортировка.
3. Сортировка просеиванием (нисходящая пузырьковая).
4. Сортировка просеиванием с бинарным поиском позиции вставки.

Алгоритмы “быстрой” сортировки:

5. Сортировка слиянием с доп. памятью $O(N)$ — рекурсивный нисходящий вариант.
6. Сортировка слиянием с доп. памятью $O(N)$ — итерационный восходящий вариант.
7. Сортировка слиянием без доп. памяти с трудоемкостью $O(N \log^2 N)$.
8. Быстрая сортировка (quicksort) с двойной рекурсией.
9. Быстрая сортировка (quicksort) с одинарной рекурсией.
10. Пирамидальная сортировка (heapsort).

“Специальные” сортировки:

11. Сортировка подсчетом для целого массива.
12. Сортировка массива целых чисел битовым аналогом быстрой сортировки.
13. Сортировка массива целых чисел байтовым radix методом с сортировкой подсчетом.
14. Сортировка массива целых чисел битовым radix методом.