

СОРТИРОВКА СЛИЯНИЕМ, БЫСТРАЯ СОРТИРОВКА, СОРТИРОВКА ПОДСЧЕТОМ, СЛИЯНИЕ УПОРЯДОЧЕННЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

1. Опишите шаблонную функцию для алгоритма сортировки слиянием для массивов.

```
template <class T>
void merge_sort(T *mas, int size);
```

Требования:

- Алгоритм нерекурсивный.
- Алгоритм должен использовать вспомогательную память размером в два раза меньше, чем размер исходного массив.
- Дополнительная память должна выделяться ровно один раз в течение всего алгоритма (команда new должна выполняться ровно один раз).
- Не забудьте предусмотреть освобождение дополнительной памяти.
- Алгоритм должен переходить к квадратичному алгоритму сортировки при малой размерности сортируемой последовательности.

2. Реализуйте алгоритм сортировки слиянием для односвязных линейных списков без использования дополнительной памяти.

```
struct elem{
    int info;
    elem*next;
};
void merge_sort_list(elem * &top);
```

Требования:

- В ходе алгоритма запрещается выделение памяти (команда new). Список должен сортироваться путем перенаправления ссылок.

3. Реализуйте шаблонную функцию для сортировки массива методом Quick_Sort.

```
template <class T>
void quick_sort(T *mas, int size);
```

Требования:

- 1) В алгоритме должен использоваться только один рекурсивный вызов.

4. Реализуйте шаблонную функцию для алгоритма оптимального слияния упорядоченных массивов разного размера: массива `mas1` размера `size1` и массива `mas2` размера `size2` ($size1 \leq size2$). Функция должна возвращать массив (назовем его `mas3`) размера `size1`, содержащего для каждого элемента массива `mas1` индекс элемента в массиве `mas2`, перед которым данный элемент должен находиться, если мы будем выписывать ответ.

```
template <class T>
int* optimal_merge(T *mas1, int size1, T *mas2, int
size2);
```

Пример:

Вход:

mas1 = (2,4,8,9,15), *size1* = 5

mas2 = (1,3,4,6,7,10,13), *size* = 7

Выход: (1,2,5,5,7)

Требования:

- Сложность алгоритма должна составлять $O(size1 \log \frac{size2}{size1})$.
- Для поиска местоположения очередного элемента массива `mas1` внутри массива `mas2` использовать алгоритм `Finger_Search`.
- Для проверки результата напишите и используйте функцию, которая выписывает результирующую упорядоченную последовательность, полученную в результате объединения последовательностей `mas1` и `mas2`, с использованием построенного массива индексов `mas3`.

5. В текстовом файле «Data_100000.txt» задана информация о людях. В каждой строке файла записана информация об одном человеке:

- Имя
- Отчество
- Фамилия,
- Дата рождения,
- Уникальный код.

В файле представлена информация о 100000 людях. Отсортировать всех людей по дате рождения, используя сортировку подсчетом в три прохода: сначала по году рождения, потом – по месяцу, потом – по дню рождения. Сложность по времени должна составлять $O(n)$, где n – количество строк в исходном текстовом файле.