

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ Н. Э. БАУМАНА

ФУНКЦИОНАЛЬНАЯ ЛОГИКА И ТЕОРИЯ
АЛГОРИТМОВ

ОТЧЕТ О ВЫПОЛНЕНИИ КОМАНДНОГО
ЗАДАНИЯ НА ТЕМУ:

“Реализация алгоритмов сортировки пузырьком и
пирамидальной сортировки на языке
программирования Си”

Выполнили: студенты группы ИУ4-33Б -

- 1) Кондратив Владимир
- 2) Салуев Евгений
- 3) Сусликов Антон
- 4) Фомичев Павел
- 5) Шадрин Юрий
- 6) Яковлев Иван

Проверил: Терехов Владимир Владимирович

Цель работы: изучить основные алгоритмы построения пирамидальной и пузырьковой сортировок. Ознакомится опытным путем с преимуществами и недостатками данных алгоритмов.

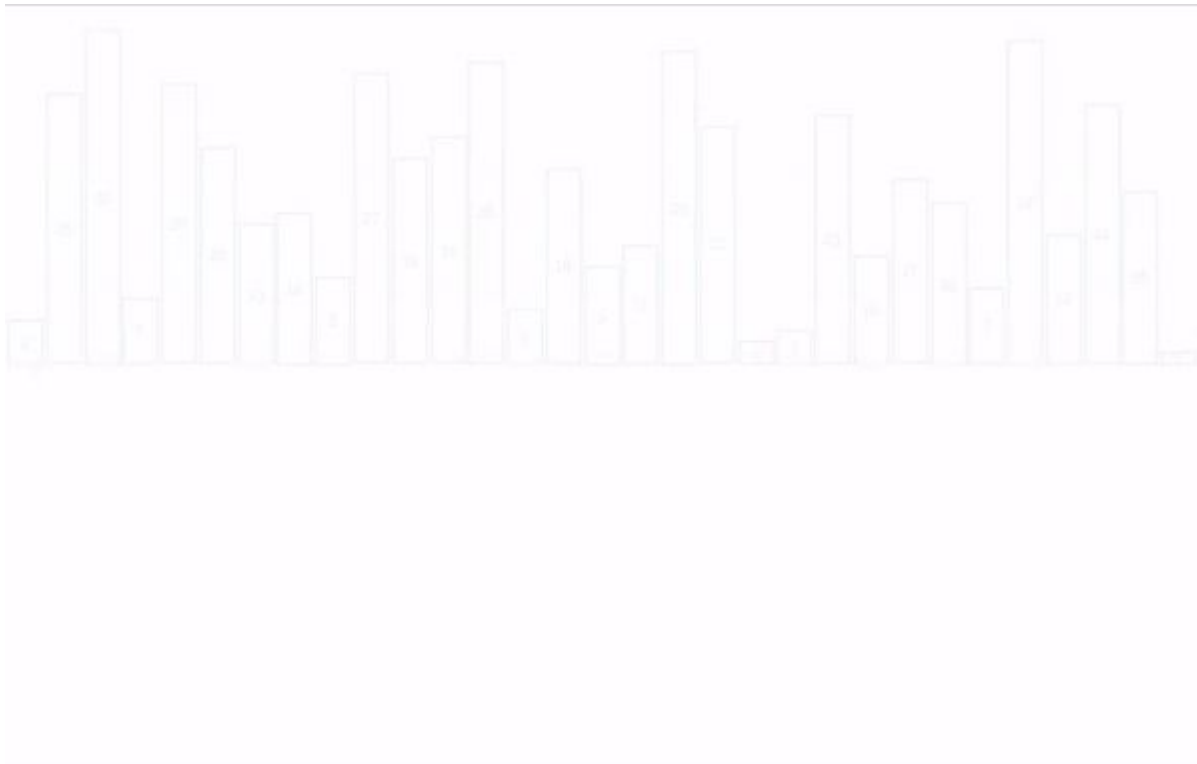
Задачи: разработать и реализовать на языке программирования Си алгоритмы пирамидальной и пузырьковой сортировок; для данных алгоритмов дать характеристику и построить структурные схемы согласно ГОСТ 19.701-90

Пирамидальная сортировка

Пирамидальная сортировка — алгоритм сортировки, работающий в худшем, в среднем и в лучшем случае (то есть гарантированно) за $\Theta(n \log n)$ операций при сортировке n элементов. Количество применяемой служебной памяти не зависит от размера массива (то есть, $O(1)$).

Алгоритм пирамидальной сортировки можно рассматривать как улучшенную версию алгоритма сортировки выбором: он делит входные данные на отсортированную и несортированную области, а затем последовательно уменьшает несортированную область, извлекая самый большой элемент и перемещая его в отсортированную область. Улучшение состоит в том, что используется бинарная куча, а не алгоритм линейного поиска, чтобы найти наибольшее значение.

Пирамидальная сортировка



Достоинства и недостатки

Достоинства:

- Имеет доказанную оценку худшего случая $O(n \log n)$.
- Требуется всего $O(n \log n)$ дополнительной памяти.

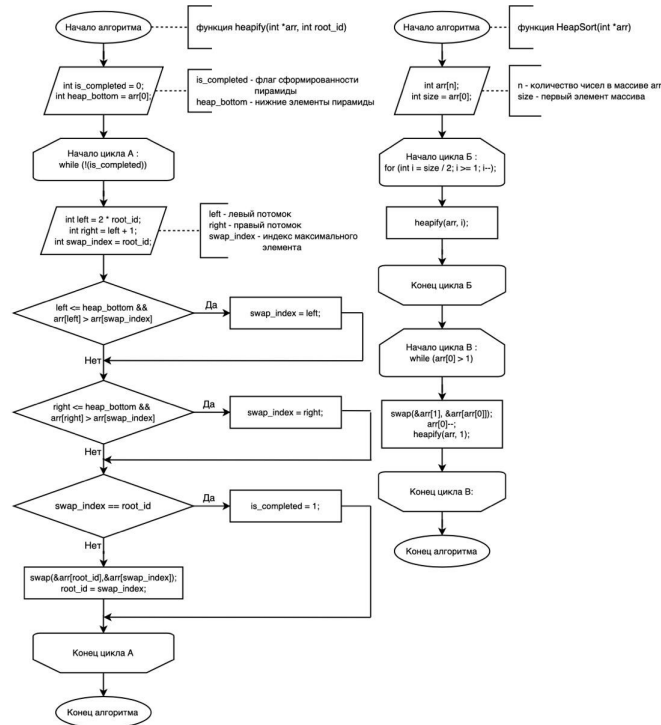
Недостатки:

- Сложен в реализации.
- Неустойчив -- для обеспечения устойчивости нужно расширять ключ.
- На почти отсортированных массивах работает столь же долго, как и на хаотических данных.
- На одном шаге выборку приходится делать хаотично по всей длине массива -- поэтому алгоритм плохо сочетается с кэшированием и подкачкой памяти.

Код программы

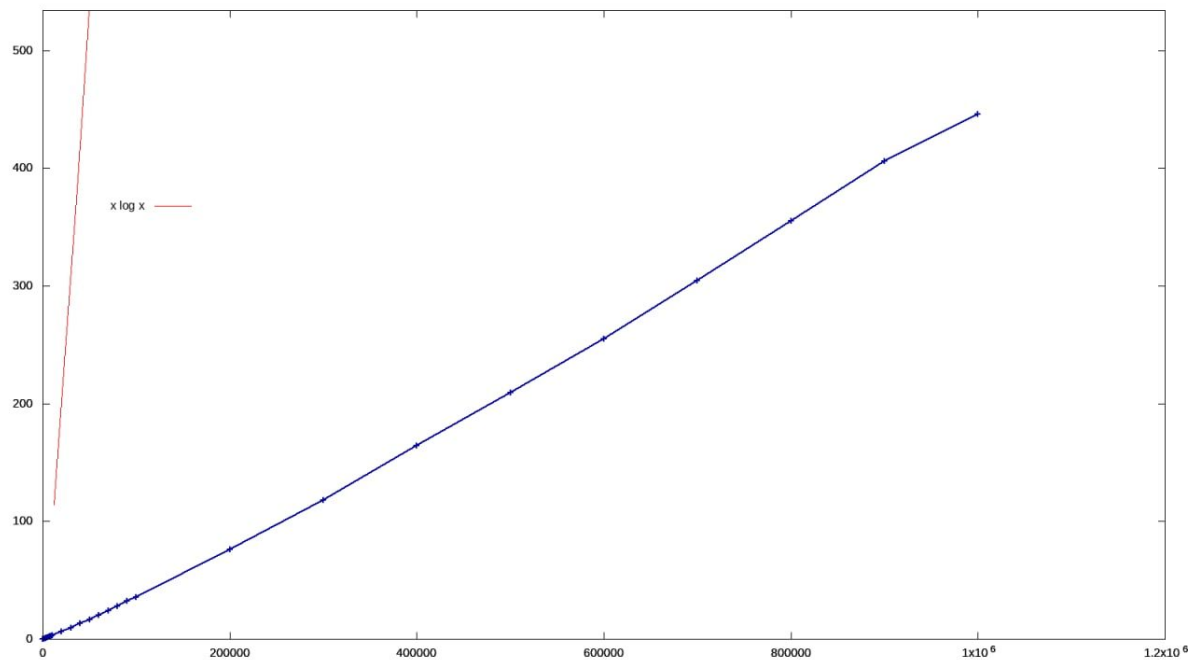
```
1  #include <stdio.h>
2
3  void heapify(int *arr, int root_id) {
4      int is_completed = 0;
5      int heap_bottom = arr[0];
6
7      while (!(is_completed)) {
8
9          int left = 2 * root_id, right = left + 1, swap_index = root_id;
10
11         if (left <= heap_bottom && arr[left] > arr[swap_index])
12             swap_index = left;
13         if (right <= heap_bottom && arr[right] > arr[swap_index])
14             swap_index = right;
15         if (swap_index == root_id)
16             is_completed = 1;
17         else {
18             swap(&arr[root_id], &arr[swap_index]);
19             root_id = swap_index;
20         }
21     }
22 }
23
24 void HeapSort(int *arr) {
25     int size = arr[0];
26     for (int i = size / 2; i >= 1; i--)
27         heapify(arr, i);
28
29     while (arr[0] > 1) {
30         swap(&arr[1], &arr[arr[0]]);
31         arr[0]--;
32         heapify(arr, 1);
33     }
34 }
```

Структурная схема алгоритма пирамидальная сортировка



Оценка сложности

Алгоритм сортировки работает в худшем, в среднем и в лучшем случае (то есть гарантированно) за $O(n \log n)$ операций при сортировке n элементов. Количество применяемой служебной памяти не зависит от размера массива (то есть, $O(1)$).

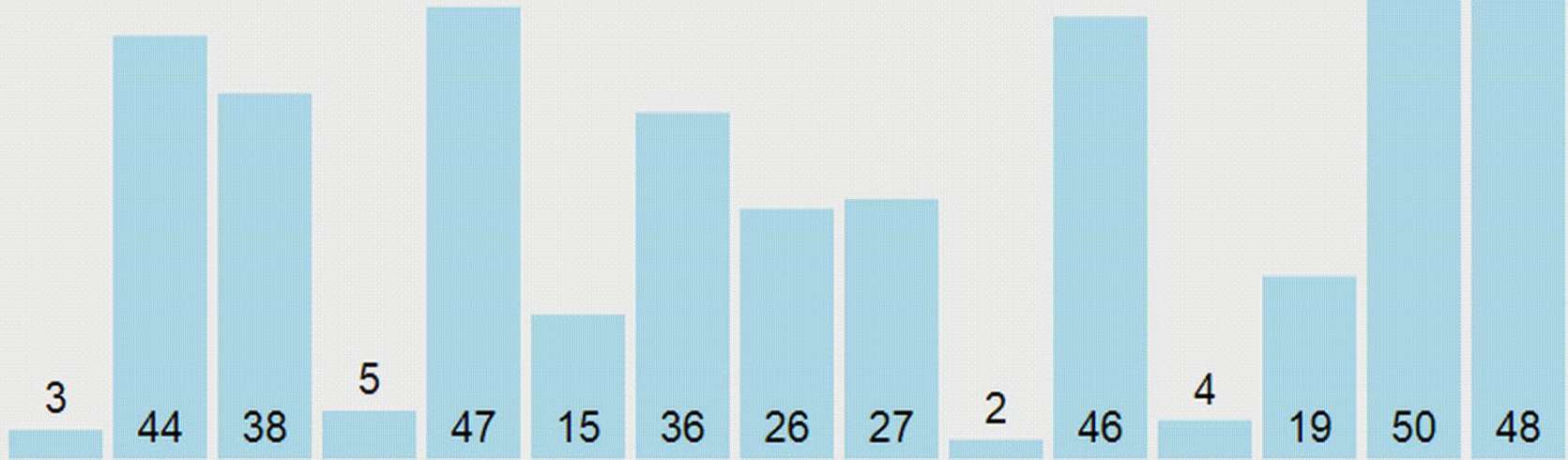


Сортировка пузырьком

Сортировка пузырьком — простой алгоритм сортировки. Для понимания и реализации этот алгоритм — простейший, но эффективен он лишь для небольших массивов. Сложность алгоритма: $O(n^2)$.

Принцип действий прост: обходим массив от начала до конца, попутно меняя местами неотсортированные соседние элементы. В результате первого прохода на последнее место «всплывёт» максимальный элемент. Теперь снова обходим неотсортированную часть массива (от первого элемента до предпоследнего) и меняем по пути неотсортированных соседей. Второй по величине элемент окажется на предпоследнем месте. Продолжая в том же духе, будем обходить всё уменьшающуюся неотсортированную часть массива, запихая найденные максимумы в конец.

Сортировка пузырьком



Достоинства и недостатки

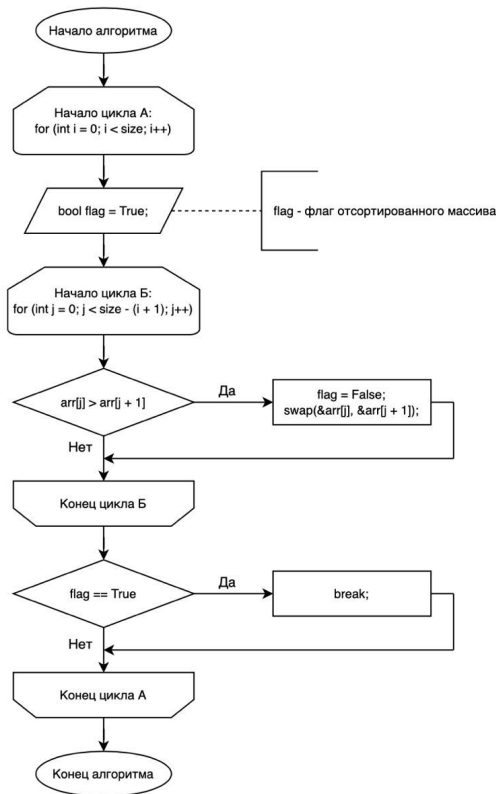
Достоинство метода: не требуется дополнительных массивов

Недостаток: время алгоритма пропорционально квадрату количества элементов
(самый медленный способ сортировки)

Код программы

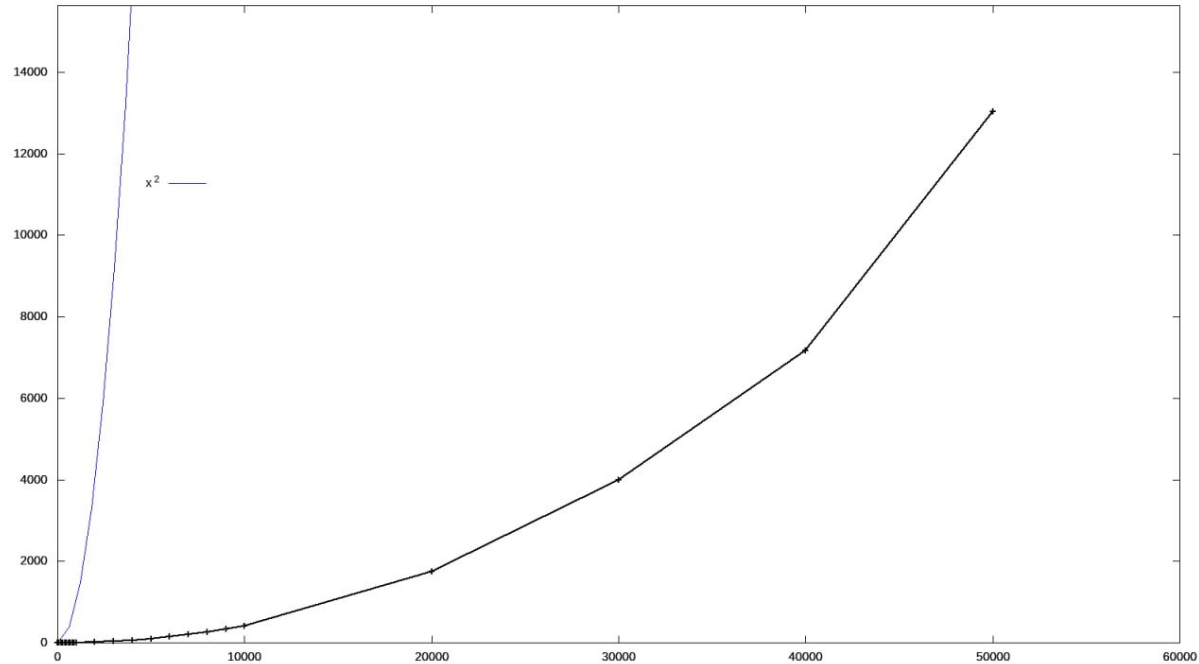
```
1  #include <stdio.h>
2
3
4  void BubbleSort(int *arr, int size) {
5      for (int i = 0; i < size; i++) {
6          bool flag = True;
7          for (int j = 0; j < size - (i + 1); j++) {
8              if (arr[j] > arr[j + 1]) {
9                  flag = False;
10                 swap(&arr[j], &arr[j + 1]);
11             }
12         }
13         if (flag == True) break;
14     }
15 }
```

Структурная схема алгоритма пирамидальная сортировка



Оценка сложности

Алгоритм сортировки работает в худшем, в среднем за $O(n^2)$, в лучшем случае за $O(n)$ операций при сортировке n элементов.



Выводы

Практическая временная сложность сортировки пузырьком получилась несколько меньше теоретической, т.к. она оптимизирована, однако даже с оптимизацией пузырьковая сортировка является более медленной по сравнению с пирамидальной.