

Цель работы - изучить применение предварительной сортировки при решении алгоритмических задач, а также сравнить эффективность различных алгоритмов обработки массивов целых чисел.

Постановка задач

В рамках семинара были рассмотрены следующие задачи:

- 1) Поиск медианы массива.
- 2) Поиск пересечения двух множеств.
- 3) Поиск минимального и максимального элементов массива.

Задача 1. Поиск медианы

Медианой множества из n чисел называется элемент с индексом $\lfloor n/2 \rfloor$ в массиве, отсортированном по возрастанию.

Для решения задачи массив предварительно сортируется, после чего возвращается элемент с индексом $\lfloor n/2 \rfloor$. Сложность алгоритма: $O(n \log n)$.

```
===== Семинар 16 =====
1) Задача 1: медиана через сортировку
2) Задача 3: пересечение множеств A и B
3) Задача 4: минимум и максимум (3 алгоритма)
0) Выход
Выберите пункт: 1

--- Задача 1: медиана через сортировку ---
Как задать массив?
  1) Ввести вручную
  2) Сгенерировать случайно
Выбор: 2
Введите размер массив: 100
Минимальное значение: 5
Максимальное значение: 10000
Массив сгенерирован.
Массив: [8794, 4256, 6560, 5702, 8507, 9702, 5046, 9612, 391, 7441, 3986, 2127, 6033, 7970, 9094, 4488, 5570, 4884, 1608, 8095, 8050, 6092, 7110, 7074, 381, 2015, 5389, 9558, 2598, 9613, 8230, 3069, 9182, 7387, 3571, 3085, 7683, 4846, 7827, 3692, 5628, 737, 8639, 3810, 1552, 3013, 3704, 8465, 1989, 4262, 2113, 1364, 2103, 7073, 7146, 9433, 9755, 4790, 2292, 1054, 3860, 522, 5112, 7105, 6988, 1717, 4304, 7606, 7396, 8190, 5382, 3555, 7134, 2880, 4222, 9027, 8575, 2086, 2425, 4320, 8672, 8079, 2033, 7061, 9240, 5747, 1541, 9908, 7122, 5219, 2735, 6696, 402, 6535, 6236, 5532, 7220, 88, 5304, 9426]
Медиана (элемент с индексом n/2 после сортировки): 5702
Сложность: O(n log n)
```

Рисунок 1 - Пример работы программы при поиске медианы массива

Задача 3. Пересечение двух множеств

Для поиска пересечения множеств А и В реализованы два алгоритма. Алгоритм грубой силы: каждый элемент множества А сравнивается со всеми элементами множества В. Сложность алгоритма: $O(n \cdot m)$.

Алгоритм с предварительной сортировкой: оба множества сортируются, после чего используется метод двух указателей. Сложность алгоритма: $O(n \log n + m \log m)$.

```
---- Задача 3: пересечение множеств А и В ----
Как задать А?
  1) Ввести вручную
  2) Сгенерировать случайно
Выбор: 2
Введите размер А: 50
Минимальное значение: 1
Максимальное значение: 100
А сгенерирован.
Как задать В?
  1) Ввести вручную
  2) Сгенерировать случайно
Выбор: 2
Введите размер В: 80
Минимальное значение: 1
Максимальное значение: 100
В сгенерирован.
А = [69, 59, 30, 44, 35, 90, 33, 78, 97, 72, 58, 76, 71, 27, 89, 98, 58, 40, 61, 3, 89, 86, 15, 37, 81, 79, 97, 36, 2, 9, 92, 78, 91, 34, 68, 25, 99, 41, 24, 48, 44, 81, 37, 4, 90, 46, 97, 7, 1, 1, 39]
В = [51, 54, 37, 21, 18, 27, 96, 38, 40, 98, 51, 92, 32, 79, 43, 34, 25, 65, 49, 72, 29, 13, 48, 21, 47, 35, 32, 51, 2, 46, 47, 59, 51, 17, 84, 67, 57, 9, 93, 9, 72, 66, 6, 80, 7, 50, 88, 77, 79, 89, 22, 97, 20, 60, 95, 30, 66, 13, 13, 15, 89, 16, 69, 99, 70, 92, 46, 86, 6, 46, 18, 69, 61, 37, 37, 98, 43, 8, 80, 54]

(a) Грубая сила ( $O(n \cdot m)$ ):
С = [69, 59, 30, 35, 97, 72, 27, 89, 98, 40, 61, 86, 15, 37, 79, 29, 92, 34, 25, 99, 48, 46, 7]

(b) Сортировка + два указателя ( $O(n \log n + m \log m)$ ):
С = [7, 15, 25, 27, 29, 30, 34, 35, 37, 40, 46, 48, 59, 61, 69, 72, 79, 86, 89, 92, 97, 98, 99]
```

Рисунок 2 — Результат выполнения алгоритмов поиска пересечения множеств

Задача 4. Поиск минимума и максимума

В работе рассмотрены три подхода к поиску минимального и максимального элементов массива:

- 1) Алгоритм грубой силы — один линейный проход по массиву.
- 2) Алгоритм с предварительной сортировкой.
- 3) Алгоритм декомпозиции (попарное сравнение).

Алгоритмы грубой силы и декомпозиции имеют линейную сложность $O(n)$, в то время как алгоритм с сортировкой имеет сложность $O(n \log n)$.

```
===== Задача 4: минимум и максимум =====
Как задать массив?
1) Ввести вручную
2) Сгенерировать случайно
Выбор: 2
Введите размер массива: 100
Минимальное значение: 4
Максимальное значение: 1000
массив сгенерирован.
Массив: [908, 491, 487, 831, 376, 42, 299, 283, 759, 814, 704, 563, 142, 561, 170, 894, 290, 145, 173, 152, 177, 898, 447, 623, 691, 892, 219, 838, 432, 175, 161, 8, 242, 384, 864, 726, 764, 2, 64, 874, 981, 177, 618, 468, 324, 729, 185, 484, 354, 988, 997, 771, 39, 755, 611, 315, 179, 126, 846, 963, 884, 11, 571, 696, 667, 747, 913, 458, 963, 975, 624, 633, 854, 998, 985, 626, 14, 7, 57, 962, 136, 323, 511, 401, 567, 463, 755, 316, 920, 900, 544, 426, 835, 724, 425, 623, 345, 185, 201, 456, 72, 417]

1) Грубая сила (1 проход): min=8, max=998 | сложность O(n)
2) Через сортировку:      min=8, max=998 | сложность O(n log n)
3) Декомпозиция (пары):   min=8, max=998 | сложность O(n)

Сравнение:
- Грубая сила и декомпозиция работают за O(n).
- Сортировка медленнее по асимптотике: O(n log n), но после сортировки могут быть полезны другие операции.

===== Семинар 16 =====
1) Задача 1: медиана через сортировку
2) Задача 3: пересечение множеств A и B
3) Задача 4: минимум и максимум (3 алгоритма)
0) Выход
Выберите пункт: 0
```

Рисунок 3 — Результат выполнения алгоритмов поиска минимума и максимума

Вывод

В ходе выполнения работы были реализованы и исследованы различные алгоритмы обработки массивов. Проведено сравнение их асимптотической сложности. Показано, что алгоритмы линейной сложности являются наиболее эффективными, а предварительная сортировка оправдана не во всех случаях.