

Readme.md

Лабораторная работа №1. Сложность алгоритмов и их оптимизация.

Вариант 1 - Бинарный поиск

1. [Реализация алгоритма](#)
2. Сложность алгоритма $O(\log n)$, память $O(\log n)$, среднее время работы на одном наборе данных : 0.55 мс
3. [Оптимизация по памяти](#) - данная оптимизация заключается в том, что вместо рекурсивного вызова функции на меньшем отрезке будет использовать цикл while, что поможет сократить использование памяти.
4. Сложность алгоритма $O(\log n)$, память $O(1)$, среднее время работы на одном наборе данных : 0.53 мс
5. [Тесты](#)
6. Сравнение реализаций

Классическая реализация :

```
class BinarySearch<T : Comparable<T>>(private val data: List<T>) {  
  
    fun findElement(element: T, fromIndex: Int = 0, toIndex: Int = data.size - 1): Int {  
        return if (data.isEmpty() || toIndex < fromIndex) {  
            -1  
        } else {  
            val pos = (fromIndex + toIndex) / 2  
            when {  
                data[pos] > element -> findElement(element, fromIndex, pos - 1)  
                data[pos] < element -> findElement(element, pos + 1, toIndex)  
                else -> pos  
            }  
        }  
    }  
}
```

Оптимизированная реализация :

```
class BinarySearchMemoryOptimization<T : Comparable<T>>(private val data: List<T>) {  
  
    fun findElement(element: T, fromIndex: Int = 0, toIndex: Int = data.size - 1): Int {  
        if (data.isEmpty() || toIndex < fromIndex) {  
            return -1  
        } else {  
            var from = fromIndex  
            var to = toIndex  
            do {  
                val pos = (from + to) / 2  
                when {  
                    data[pos] > element -> to = pos - 1  
                    data[pos] < element -> from = pos + 1  
                    data[pos] == element -> return pos  
                }  
            } while (to >= from)  
            return -1  
        }  
    }  
}
```

Вывод

Оптимизация по памяти уменьшила количество потребляемых ресурсов алгоритма, немного увеличила размер кода, а также незначительно повысила скорость работы.

[Отчет в формате pdf](#)