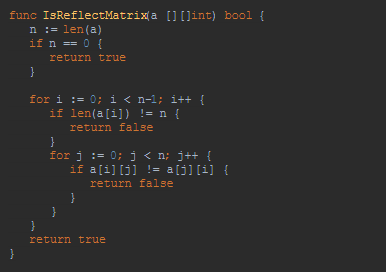
**Алгоритм №1**

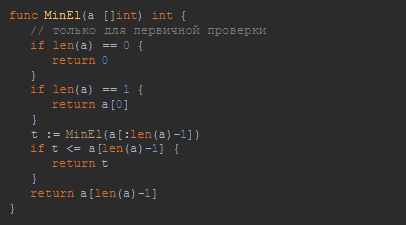
Представлен алгоритм для определения “зеркальности” матрицы размером n на n. Вычислите его сложность, используя изученные формулы, в качестве размера входных данных примите n.



|  |  |
| --- | --- |
| 1. Размер входных данных | Само число n |
| 1. Определите основную операцию алгоритма | Сравнение |
| 1. Проверьте, зависит ли число выполняемых основных операций только от размера входных данных | Нет. Кол-во операций будет зависеть не только от общего числа n элементов в массиве, но и от длины этих элементов (массивов), а также есть ли различные элементы относительно главной диагонали матрицы. |
| 1. Составьте рекуррентное уравнение, выражающее количество выполняемых основных операций алгоритма, и укажите соответствующие начальные условия. |  |
| 1. Найдите решение рекуррентного уравнения или, если это невозможно, определите хотя бы его порядок роста. |  |

**Алгоритм №2**

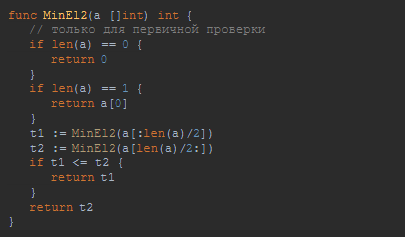
Представлен рекурсивный алгоритм для поиска наименьшего элемента слайса. Вычислите его сложность, используя изученные формулы.



|  |  |
| --- | --- |
| 1. Размер входных данных | Само число n |
| 1. Определите основную операцию алгоритма | Сравнение |
| 1. Проверьте, зависит ли число выполняемых основных операций только от размера входных данных | Да |
| 1. Составьте рекуррентное уравнение, выражающее количество выполняемых основных операций алгоритма, и укажите соответствующие начальные условия. |  |
| 1. Найдите решение рекуррентного уравнения или, если это невозможно, определите хотя бы его порядок роста. | *…* |

**Алгоритм №3**

Представлен другой рекурсивный алгоритм для поиска наименьшего элемента слайса. Вычислите его сложность, используя изученные формулы. Сравните эффективность алгоритма с предыдущим вариантом. Попробуйте увеличить скорость выполнения функции, используя инструменты Го. Как изменится сложность алгоритма при увеличении скорости?



|  |  |
| --- | --- |
| 1. Размер входных данных | k = log2n |
| 1. Определите основную операцию алгоритма | Сравнение |
| 1. Проверьте, зависит ли число выполняемых основных операций только от размера входных данных | Да |
| 1. Составьте рекуррентное уравнение, выражающее количество выполняемых основных операций алгоритма, и укажите соответствующие начальные условия. | Решение с использованием мастер-теоремы |
| 1. Найдите решение рекуррентного уравнения или, если это невозможно, определите хотя бы его порядок роста. | *, где:*   * *a = 2 – количество подзадач исходной задачи* * *b = 2 (n/b – размер каждой из подзадач)* * *f(n) = O(1) – время, затрачиваемое на разбиение задачи и объединение результатов подзадач*   Сравним f(n) c  Т.е. f(n) растет медленнее. Поэтому, согласно теореме: |

Увеличение времени выполнения можно достигнуть за счет использования горутин (см. код в следующем файле main.go). Таким образом, сложность будет O(log n)