```
MaxS (S,n; Max)
Max <-- S[1]
For i <-- 2 to n
if Max < S[i]
then Max <-- S[i]
end for
return Max
End
```

Данный алгоритм является количественно-параметрическим, поэтому для фиксированной размерности исходных данных необходимо проводить анализ для худшего, лучшего и среднего случая.

А). Худший случай

Максимальное количество переприсваиваний максимума (на каждом проходе цикла) будет в том случае, если элементы массива отсортированы по возрастанию. Трудоемкость алгоритма в этом случае равна:

$F(n)=1+1+1+ (n-1) (3+2+2)=7 n - 4 = O(n)$.

Б) Лучший случай

Минимальное количество переприсваивания максимума (ни одного на каждом проходе цикла) будет в том случае, если максимальный элемент расположен на первом месте в массиве. Трудоемкость алгоритма в этом случае равна:

$F(n)=1+1+1+ (n-1) (3+2)=5 n - 2 = O(n)$.

В) Средний случай

Алгоритм поиска максимума последовательно перебирает элементы массива, сравнивая текущий элемент массива с текущим значением максимума. На очередном шаге, когда просматривается к-ый элемент массива, переприсваивание максимума произойдет, если в подмассиве из первых к элементов максимальным элементом является последний. Очевидно, что в случае равномерного распределения исходных данных, вероятность того, что максимальный из к элементов расположен в определенной (последней) позиции равна 1/к. Тогда в массиве из n элементов общее количество операций переприсваивания максимума определяется как:

$$\sum_{i=1}^{N} 1/i = Hn \approx Ln(N) + \gamma, \quad \gamma \approx 0{,}57$$

Величина Hn называется n-ым гармоническим числом. Таким образом, точное значение (математическое ожидание) среднего количества операций присваивания в алгоритме поиска максимума в массиве из n элементов определяется величиной H_n (на бесконечности количества испытаний), тогда:

$F(n)=1 + (n-1)(3+2) + 2(Ln(n) + \gamma) = 5n + 2 Ln(n) - 4 + 2\gamma = O(n)$.