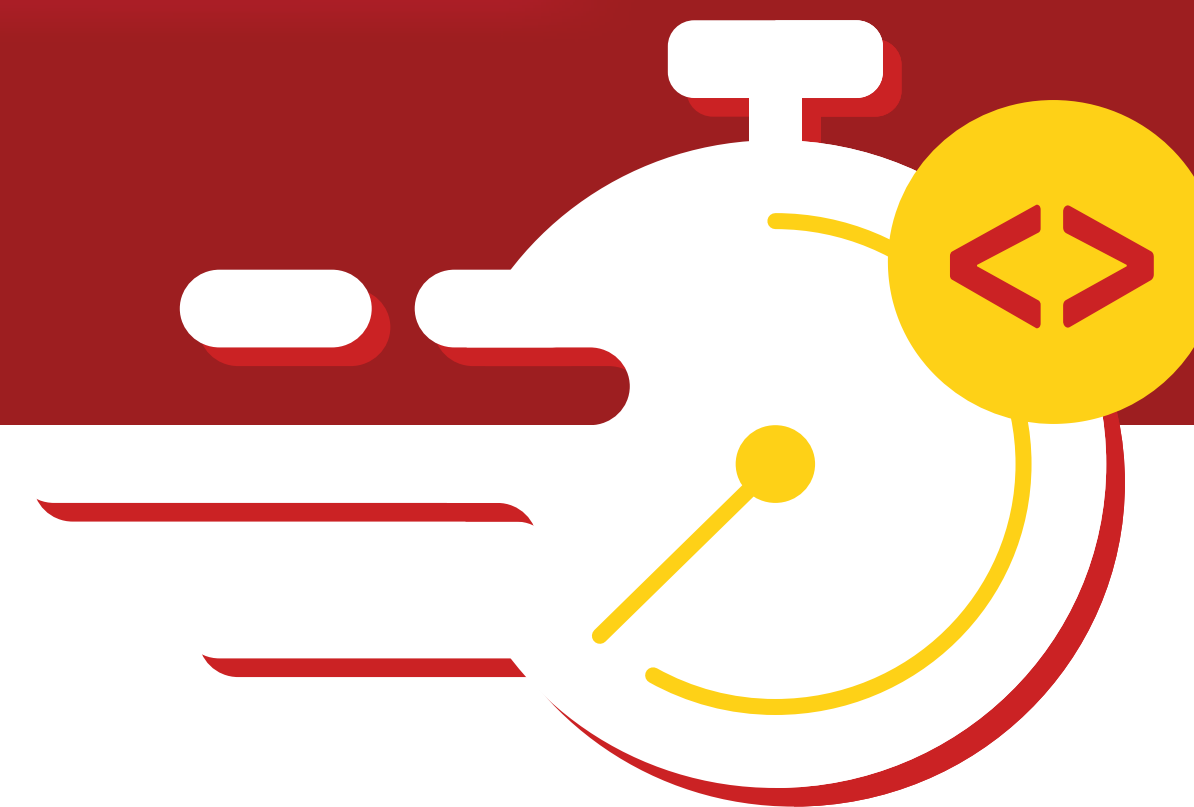
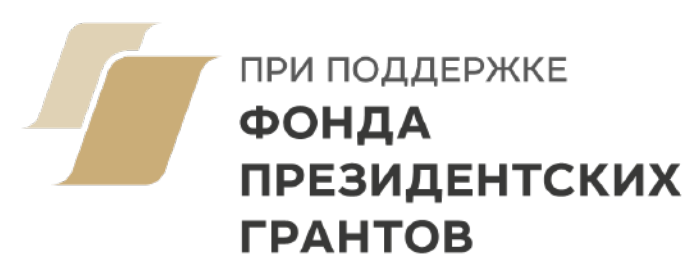


Линейные алгоритмы

Урок 1.2



В этом уроке_

- Практические примеры линейных алгоритмов

Линейные алгоритмы: задача о максимуме_

- Дано целое положительное число n , не превосходящее 10^7 , а затем – n целых чисел, разделенных пробелами и не превосходящих по модулю 10^9 .
- Требуется напечатать одно число — максимум среди введенных n чисел
- Например, для ввода
 - 5
 - 1 7 5 2 3требуется вывести число 7

Поиск максимума_

Особенности «олимпиадного» кода:

- Использование `using namespace std`
- Использование глобальных переменных и массивов

```
1  #include <iostream>
2
3  using namespace std;
4
5  const int MAXN = 10000500;
6  const int MAXVALUE = 1000000500;
7  int n;
8  int a[MAXN];
9
10 int main() {
11     ios_base::sync_with_stdio(0);
12     cin.tie(0);
13
14     cin >> n;
15     for (int i = 0; i < n; ++i)
16         cin >> a[i];
17
18     int maxa = -MAXVALUE - 1;
19     for (int i = 0; i < n; ++i)
20         if (maxa < a[i]) {
21             maxa = a[i];
22         }
23
24     cout << max2a << "\n";
25     return 0;
26 }
```

Поиск максимума_

- Строчки 11-12 — для ускорения считывания cin-ом
- НЕ печатать на экран фразы в духе «Введите число n:» перед считыванием

```
1  #include <iostream>
2
3  using namespace std;
4
5  const int MAXN = 10000500;
6  const int MAXVALUE = 1000000500;
7  int n;
8  int a[MAXN];
9
10 int main() {
11     ios_base::sync_with_stdio(0);
12     cin.tie(0);
13
14     cin >> n;
15     for (int i = 0; i < n; ++i)
16         cin >> a[i];
17
18     int maxa = -MAXVALUE - 1;
19     for (int i = 0; i < n; ++i)
20         if (maxa < a[i]) {
21             maxa = a[i];
22         }
23
24     cout << max2a << "\n";
25     return 0;
26 }
```

Поиск максимума: асимптотика_

- Строки 14, 18, 23 работают за $O(1)$
- If в строке 20 выполняется $n+1 \leq 2n = O(n)$ раз; присваивание в строке 21 – не более чем $O(n)$
- Цикл for в строках 15-16 работает также за $O(n)$
- Итоговая сложность – $O(1) + O(n) + O(n) = O(n)$

```
1  #include <iostream>
2
3  using namespace std;
4
5  const int MAXN = 10000500;
6  const int MAXVALUE = 1000000500;
7  int n;
8  int a[MAXN];
9
10 int main() {
11     ios_base::sync_with_stdio(0);
12     cin.tie(0);
13
14     cin >> n;
15     for (int i = 0; i < n; ++i)
16         cin >> a[i];
17
18     int maxa = -MAXVALUE - 1;
19     for (int i = 0; i < n; ++i)
20         if (maxa < a[i]) {
21             maxa = a[i];
22         }
23     cout << max2a << "\n";
24     return 0;
25 }
```

Линейные алгоритмы:

задача о втором максимуме_

- Дано целое положительное число n , не меньшее 2 и не превосходящее 10^7 , а затем – n целых чисел, разделенных пробелами и не превосходящих по модулю 10^9
- Требуется напечатать одно число – второй максимум среди введенных n чисел, т.е. такое число, что если упорядочить числа по неубыванию, то напечатанное число окажется предпоследним в списке
- Например, для ввода
 - 5
 - 1 7 5 2 3требуется вывести число 5

Поиск второго максимума_

Напрашивающийся метод:

- Второй максимум — это максимум, найденный без учета максимального элемента
- Полученный алгоритм работает за $O(n)$
- Найдите ошибку!

```
1  #include <iostream>
2
3  using namespace std;
4
5  const int MAXN = 10000500;
6  const int MAXVALUE = 1000000500;
7  int n;
8  int a[MAXN];
9
10 int main() {
11     // ... считывание ...
12
13     // поиск максимума
14     int maxa = -MAXVALUE;
15     for (int i = 0; i < n; ++i)
16         if (maxa < a[i])
17             maxa = a[i];
18
19     // поиск второго максимума
20     int max2a = -MAXVALUE;
21     for (int i = 0; i < n; ++i)
22         if (a[i] != maxa)
23             if (max2a < a[i])
24                 max2a = a[i];
25
26     cout << max2a << "\n";
27     return 0;
28 }
```


Поиск второго максимума_

- Ошибка: второй максимум вычислялся без учета не максимального элемента, но всех элементов, являющихся максимальным

```
1  #include <iostream>
2
3  using namespace std;
4
5  const int MAXN = 10000500;
6  const int MAXVALUE = 1000000500;
7  int n;
8  int a[MAXN];
9
10 int main() {
11     // ... считывание ...
12
13     // поиск максимума
14     int maxa = -MAXVALUE;
15     for (int i = 0; i < n; ++i)
16         if (maxa < a[i])
17             maxa = a[i];
18
19     // поиск второго максимума
20     int max2a = -MAXVALUE;
21     for (int i = 0; i < n; ++i)
22         if (a[i] != maxa)
23             if (max2a < a[i])
24                 max2a = a[i];
25
26     cout << max2a << "\n";
27     return 0;
28 }
```

Поиск второго максимума_

- Чтобы избежать такого эффекта, будем хранить не только максимум, но и индекс одного из его вхождений

```
1  #include <iostream>
2
3  using namespace std;
4
5  const int MAXN = 10000500;
6  const int MAXVALUE = 1000000500;
7  const int NOINDEX = -1;
8  int n;
9  int a[MAXN];
10
11 int main(){
12     // ... считывание ...
13
14     int maxa = -MAXVALUE;
15     int indmaxa = NOINDEX;
16     for (int i = 0; i < n; ++i)
17         if (maxa < a[i]) {
18             maxa = a[i];
19             indmaxa = i;
20         }
21
22     int max2a = -MAXVALUE;
23     for (int i = 0; i < n; ++i)
24         if (i != indmaxa)
25             if (max2a < a[i])
26                 max2a = a[i];
27
28     cout << max2a << "\n";
29     return 0;
30 }
```