

ПОИСК МАКСИМАЛЬНОГО И МИНИМАЛЬНОГО ЭЛЕМЕНТА МАССИВА ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ НА УРОКАХ ИНФОРМАТИКИ

Волгин Павел Михайлович, pavel.niglov@yandex.ru

МОУ «Средняя Общеобразовательная Школа №20», город Воскресенск

Искусство учителя состоит в том, чтобы рассказать сложные вещи простыми словами. Учебный предмет «Информатика» полон сложных вещей, о которых без специальной подготовки невозможно рассказать детям. Сложной вещью в информатике можно назвать массив. Массив – это сложный тип данных в программировании. В школьном курсе информатики массивы изучают уже при завершении обучения программированию. В учебнике Семакина И. Г. за 9 класс тема «Массивы» рассматриваются во второй главе в параграфе 19, а обработка массива – в параграфе 20.

Еще до операций с массивами, мы стараемся представить этот сложный тип данных на простых примерах. Говорим о том, что массив – это улица, у которой есть имя, а элементами являются дома. У каждого дома есть свой номер. Все дома расставлены по порядку на улице. У детей сразу возникает ассоциация с реальными объектами окружающей среды, поэтому само понятие массива дается им довольно легко. (Упорядоченный однотипный набор элементов – упорядоченный набор однотипных домов)

Проблема в понимании массива возникает в случае, когда учитель пытается рассказать об операциях с массивами, а именно нахождения максимального или минимального элемента. Как правило, поиск максимального элемента в массиве мы описываем в виде арифметического цикла:

```
max:=a[1];
```

```
for i:=2 to N do
```

```
    if a[i]>max then max:=a[i];
```

Проблема в понимании данного поиска заключается в строчке с условием:

```
if a[i]>max then max:=a[i];
```

Если говорить простыми словами, то это условие можно интерпретировать вот так: «если что-то больше максимума, то максимуму присвоим что-то».

И вот на этом моменте у многих детей возникают логичные вопросы. Что значит больше максимума? Разве может быть что-то больше максимума? Это условие никогда не выполнится, потому что ни один элемент не может быть больше максимального элемента. Дети забывают про строчку, которую они описали до цикла.

```
max:=a[1];
```

Переменной *max* мы присвоили значение первого элемента массива, а не самого максимального. Как бороться с этим непониманием? Можно построить трассировочную таблицу и посмотреть, как ведет себя переменная *max*. Можно рассмотреть несколько одномерных целочисленных массивов и прогнать их через алгоритм поиска максимального элемента. Но все это будет бесполезно, если ребенок не понимает, как именно работает этот поиск.

Для понимания поиска можно использовать метод «соревнования». При объяснении понятия массива мы использовали ассоциацию с предметами окружающей среды (массив – улица, элементы – дома), то почему бы при объяснении поиска максимального элемента не рассмотреть такую же ассоциацию?

Рассмотрим соревнования по рыболовству. Пусть рыбаки – это элементы массива. Судье чемпионата по рыболовству нужно определить чемпиона. Чемпион – это тот рыбак, который поймал больше всего рыб (максимальный элемент массива). К судье подходит первый рыбак, у которого *M* рыб. Этот рыбак садится в кресло победителя, так как он единственный, кого проверил судья. Кресло победителя – это переменная *max*. Рыбак под номером один поймал *M* рыб, следовательно, *max:=a[1]*.

Далее, начиная со второго рыбака, и до последнего, судья принимает улов последующих рыбаков и сравнивает их с тем рыбаком, кто в данный момент сидит в кресле победителя. Если второй рыбак поймал больше рыб, чем тот, кто сидит на месте победителя, то теперь этот участник садится на его место. Если же второй рыбак поймал меньше рыб, то участник под номером один остается сидеть в кресле. Следовательно, кресло победителя и будет определять чемпиона турнира. Кресло победителя каждый раз переприсваивает новое значение в том случае, когда последующий рыбак поймал больше рыб, чем «предварительный» победитель. Значит, переменная max – это кресло победителя, а рыбак с номером i – это элемент массива $a[i]$; значение элемента массива – это количество пойманных рыб. Теперь, сформулируем условие. Если количество рыб i -того рыбака больше количества рыб предварительного чемпиона, то предварительным чемпионом становится этот i -тый рыбак. Меняя понятия окружающей среды на объекты в программировании, получаем: если значение i -того элемента массива больше переменной max , то переменной max присваиваем значение i -того элемента массива.

Теперь получаем понятное описание:

$max := a[1];$ {предварительный чемпион = первый рыбак}

for $i := 2$ to N do

 if $a[i] > max$ then $max := a[i];$ {если i -тый рыбак поймал больше рыб, чем предварительный чемпион, то предварительный чемпион – это i -тый рыбак}

Аналогично можно показать и поиск минимального элемента. Только теперь это не соревнования по рыболовству, а соревнования по биатлону.

Кресло чемпиона – самый быстрый биатлонист. Получаем:

$min := a[1];$ {предварительный чемпион = первый финишируемый биатлонист}

for $i := 2$ to N do

if a[i]<min then min:=a[i]; {если i-тый биатлонист показал время меньше, чем предварительный чемпион, то предварительный чемпион – это i-тый биатлонист}

Аналогии с окружающей средой всегда помогают понять сложные вещи. Применять данный метод при объяснении программирования можно не только для поиска максимального и минимального элемента одномерного массива, но и для других задач на массивы. Естественно, что данный метод подходит только при начальном знакомстве с поиском минимального и максимального элемента. После решения нескольких задач, обучающиеся должны овладеть алгоритмом не только на базе житейского уровня, но и в объектах программирования.

Литература

1. Сайт учителя информатики Сиденко Андрея Григорьевича. [Электронный ресурс] – режим доступа: <https://agsidenko.ru>. (Дата обращения: 07.03.2020)
2. Сайт учителя информатика Полякова Константина Юрьевича [Электронный ресурс] – режим доступа: <http://kpolyakov.spb.ru>. (Дата обращения: 01.03.2020)
3. Захаров Т. Б. Общая методика обучения информатики / Т. Б. Захаров – М.: Прометей, 2016. – 302 с.