Ткаченко Кирилл Станиславович
Специалист по учебно-методической работе 1-й категории, аспирант, ассистент Севастопольского государственного университета.

Программная реализация нахождения решения головоломки «Судоку» на школьном алгоритмическом языке

В статье описывается реализация алгоритма полного перебора с возвратом. В качестве примера рассматривается логическая головоломка «Судоку». Приводятся готовая программа на школьном алгоритмическом языке и результаты её работы.

Решение логических головоломок и задач может сыграть важную роль в формировании мировоззрения и грамотности в области информатики и ИКТ. Поэтому особенно важно обратить внимание обучающихся на необходимость детальной проработки типовых алгоритмов, пригодных для решения как широкого класса задач, так и головоломок. К таким алгоритмам относятся, в том числе, и рекурсивные алгоритмы поиска с возвратом [1].

Эти алгоритмы поиска хорошо зарекомендовали себя для обеспечения поиска решений во многих прикладных областях. Для воссоздания общей схемы можно опереться на решение широко известной и распространённой логической головоломки «Судоку» [2]. Моделью такой головоломки являются обобщенные латинские квадраты. Для нахождения решения необходимо заполнить числами пустые ячейки. Числа не могут повторяться как в малых квадратах

игрового поля головоломки, так и по горизонталям и вертикалям большого квадрата.

Программных реализаций, в том числе доступных на уровне исходного кода, для решений этой головоломки достаточно много. Можно отметить опубликованные реализации на языках Си, Java, Паскаль [3] и на встроенном языке 1С [4]. Но не всегда для изучающих информатику эти решения являются прозрачными. Поэтому может потребоваться построить программный код на школьном алгоритмическом языке.

Целью настоящей работы является разработка программы на школьном алгоритмическом языке для нахождения решения головоломки «Судоку».

Программа разрабатывается с рядом упрощений, не влияющих при этом на изложение. В частности, исходные данные хранятся непосредственно в исходном коде программы, а также отсутствует обработка ситуации, когда у головоломки нет решения.

Эта программа начинается с объявления переменных:

```
цел N = 9
цел таб Судоку[1:N, 1:N]
цел СчЗамен
лит таб ИсхДанные[1:N]
```

В целочисленной переменной **N** находится размер стороны большого квадрата 9, в двумерном массиве целого типа **Судоку** – состояние игрового поля с заполненными и пустыми полями. Целочисленная переменная **СчЗамен** будет инкрементироваться по факту произведения замены. Исходные данные, задающиеся в исходном тексте, располагаются в строковом одномерном массиве **ИсхДанные**.

Головная подпрограмма находится в процедуре **РешениеСудоку**:

```
алг РешениеСудоку
нач
        лог Рез
        ИнициализацияПеременных
        СтрокиВСудоку
        ПечатьСудоку
        Рез := Решить
        ПечатьСудоку
        вывод "Число замен: ", СчЗамен, нс
кон
```

Определённая переменная **Рез** нужна для сохранения состояния решения головоломки, и впоследствии программа может быть расширена путем добавления необходимой обработки ситуации. Происходит подряд вызов подпрограмм **ИнициализацияПеременных**, **СтрокиВСудоку**, **ПечатьСудоку**, **Решить**, **ПечатьСудоку**, а затем отображение числа произведённых замен.

Инициализация переменных происходит в процедуре **ИнициализацияПеременных**:

```
алг ИнициализацияПеременных
нач
     ИсхДанные[1] := "000006201"
     ИсхДанные[2] := "070510060"
```

```
        ИсхДанные[3] := "006080059"
        ИсхДанные[4] := "120000000"
        ИсхДанные[5] := "080940500"
        ИсхДанные[6] := "590000000"
        ИсхДанные[7] := "001050073"
        ИсхДанные[8] := "030460080"
        ИсхДанные[9] := "000003604"
        СчЗамен := 0
   кон
```

В этой подпрограмме одномерный массив строк **ИсхДанные** явно заполняется начальным содержимым полей головоломки, а **СчЗамен** приобретает значение 0.

Преобразование входных данных во внутреннее представление производится в процедуре **СтрокиВСудоку**:

```
алг СтрокиВСудоку
нач
    цел ТекСтрок, ТекСтолб
    нц для ТекСтрок от 1 до N
          нц для ТекСтолб от 1 до N
                Судоку[ТекСтрок, ТекСтолб] :=
Цел(ИсхДанные[ТекСтрок, ТекСтолб], 0)
          кц
    кц
кон
```

Целочисленные переменные **ТекСтрок** и **ТекСтолб** являются счётчиками для циклов с параметрами, в них располагаются, соответственно, номер текущей рассматриваемой строки игрового поля и номер текущего рассматриваемого столбца. Проходя по всем строкам и по всем столбцам, в ячейку игрового поля **Судоку** заносится целочисленная величина из соответствующей строки массива **ИсхДанные** и символа по номеру в строке.

Отображение состояния клеток игрового поля производится в процедуре **ПечатьСудоку**:

```
алг ПечатьСудоку
нач
    цел ТекСтрок, ТекСтолб
    вывод "+---+---+---+", нс
    нц для ТекСтрок от 1 до N
          нц для ТекСтолб от 1 до N
                если mod(ТекСтолб, 3) = 1
                      то
                            вывод "|"
                все
                вывод Судоку[ТекСтрок, ТекСтолб]
          кц
          вывод "|", нс
          если mod(ТекСтрок, 3) = 0
```

```
                то
                        вывод "+---+---+---+", нс
            все
    кц
    вывод нс
кон
```

Целочисленные переменные **ТекСтрок** и **ТекСтолб** вновь являются счётчиками для циклов с параметрами для номера текущей рассматриваемой строки игрового поля и номера текущего рассматриваемого столбца.

Большую часть программного кода занимает наиболее сложная функция **Решить**, производящая поиск решения. Эта функция возвращает значение **Да**, когда решение найдено, и **Нет** в противном случае. Функция начинается с определения переменных и задания их начальных значений:

```
алг лог Решить
нач
    лог таб ЦифраВСудоку[0:9]
    цел НСтроки, НСтолбца
    лог НетНулей
    цел ТекСтрок, ТекСтолб, ТекЦифра
    цел НачСтрок, НачСтолб, КонСтрок, КонСтолб
    лог Рез
    НСтроки := 0
    НСтолбца := 0
    НетНулей := да
```

Одномерный массив логических значений **ЦифраВСудоку** отмечает в себе факт присутствия цифры и, соответственно, невозможность ее использования. **НСтроки**, **НСтолбца** – номера строки и столбца с нулевым элементом на поле, начальные значения 0. **НетНулей** – отсутствие нулей на поле, начальное значение **Да**. **ТекСтрок**, **ТекСтолб**, **ТекЦифра** – счётчики циклов, хранящие в себе номер рассматриваемой строки, столбца и цифры. **НачСтрок**, **НачСтолб**, **КонСтрок**, **КонСтолб** – граничные координаты малого квадрата, в котором находится нулевой элемент. **Рез** – результат выполнения функции.

С применением стандартного алгоритма поиска первого вхождения элемента в двумерном массиве производится нахождение крайнего слева сверху нулевого элемента и запоминание в переменных **НСтроки**, **НСтолбца** его координат:

```
    ТекСтрок := 1
    нц пока НетНулей и (ТекСтрок <= N)
          ТекСтолб := 1
          нц пока НетНулей и (ТекСтолб <= N)
                если Судоку[ТекСтрок, ТекСтолб] = 0
                      то
                            НетНулей := нет
```

```
                        НСтроки := ТекСтрок
                        НСтолбца := ТекСтолб
                все
                ТекСтолб := ТекСтолб + 1
        кц
        ТекСтрок := ТекСтрок + 1
кц
```

Если нулевой элемент не найден, то решение получено и можно успешно завершить:

```
если НетНулей
        то
                знач := да
                выход
все
```

Иначе инициализируется массив **ЦифраВСудоку** и рассчитываются границы квадрата, которому принадлежит нулевой элемент:

```
нц для ТекЦифра от 0 до 9
      ЦифраВСудоку[ТекЦифра] := нет
кц
НачСтрок := div(НСтроки - 1, 3) * 3 + 1
НачСтолб := div(НСтолбца - 1, 3) * 3 + 1
КонСтрок := НачСтрок + 2
КонСтолб := НачСтолб + 2
```

После этого отмечаются все использованные цифры на пересечении строки и столбца с нулевым элементом, а также все цифры, входящие в малый квадрат с нулевым элементом:

```
нц для ТекСтрок от 1 до N
      ЦифраВСудоку[Судоку[НСтроки, ТекСтрок]] := да
      ЦифраВСудоку[Судоку[ТекСтрок, НСтолбца]] := да
кц
нц для ТекСтрок от НачСтрок до КонСтрок
      нц для ТекСтолб от НачСтолб до КонСтолб
            ЦифраВСудоку[Судоку[ТекСтрок, ТекСтолб]] := да
      кц
кц
```

Наконец, когда был завершён подготовительный этап, начинается собственно рекурсивный поиск в возвратом. Для всех цифр от 1 до 9 происходит замена нулевого элемента на цифру, если она не использована, и процесс поиска решения рекурсивно повторяется для изменённого поля. В случае успешного решения происходит успешный возврат, иначе замена цифры отменяется и процесс поиска продолжается:

```
    Рез := нет
    ТекЦифра := 1
    нц пока не Рез и (ТекЦифра <= 9)
          если не ЦифраВСудоку[ТекЦифра]
                то
                      СчЗамен := СчЗамен + 1
                      Судоку[НСтроки, НСтолбца] := ТекЦифра
                      если Решить
                            то
                                  Рез := да
                                  выход
                      все
                      Судоку[НСтроки, НСтолбца] := 0
          все
          ТекЦифра := ТекЦифра + 1
    кц
```

Происходит возврат из функции:

```
        знач := Рез
кон
```

Полный исходный текст программы приводится в листинге 1, а результаты работы – в листинге 2.

Листинг 1. Полный исходный текст программы

```
цел N = 9
цел таб Судоку[1:N, 1:N]
цел СчЗамен
лит таб ИсхДанные[1:N]
алг РешениеСудоку
нач
    лог Рез
    ИнициализацияПеременных
    СтрокиВСудоку
    ПечатьСудоку
    Рез := Решить
    ПечатьСудоку
    вывод "Число замен: ", СчЗамен, нс
кон
алг ИнициализацияПеременных
нач
    ИсхДанные[1] := "000006201"
    ИсхДанные[2] := "070510060"
    ИсхДанные[3] := "006080059"
    ИсхДанные[4] := "120000000"
    ИсхДанные[5] := "080940500"
    ИсхДанные[6] := "590000000"
```

```
    ИсхДанные[7] := "001050073"
    ИсхДанные[8] := "030460080"
    ИсхДанные[9] := "000003604"
    СчЗамен := 0
кон
алг СтрокиВСудоку
нач
    цел ТекСтрок, ТекСтолб
    нц для ТекСтрок от 1 до N
          нц для ТекСтолб от 1 до N
                Судоку[ТекСтрок, ТекСтолб] :=
Цел(ИсхДанные[ТекСтрок, ТекСтолб], 0)
          кц
    кц
кон
алг ПечатьСудоку
нач
    цел ТекСтрок, ТекСтолб
    вывод "+---+---+---+", нс
    нц для ТекСтрок от 1 до N
          нц для ТекСтолб от 1 до N
                если mod(ТекСтолб, 3) = 1
                      то
                            вывод "|"
                все
                вывод Судоку[ТекСтрок, ТекСтолб]
          кц
          вывод "|", нс
          если mod(ТекСтрок, 3) = 0
                то
                      вывод "+---+---+---+", нс
          все
    кц
    вывод нс
кон
алг лог Решить
нач
    лог таб ЦифраВСудоку[0:9]
    цел НСтроки, НСтолбца
    лог НетНулей
    цел ТекСтрок, ТекСтолб, ТекЦифра
    цел НачСтрок, НачСтолб, КонСтрок, КонСтолб
    лог Рез
    НСтроки := 0
    НСтолбца := 0
    НетНулей := да
    ТекСтрок := 1
    нц пока НетНулей и (ТекСтрок <= N)
          ТекСтолб := 1
          нц пока НетНулей и (ТекСтолб <= N)
                если Судоку[ТекСтрок, ТекСтолб] = 0
```

```
                        то
                              НетНулей := нет
                              НСтроки := ТекСтрок
                              НСтолбца := ТекСтолб
                  все
                  ТекСтолб := ТекСтолб + 1
            кц
            ТекСтрок := ТекСтрок + 1
      кц
      если НетНулей
            то
                  знач := да
                  выход
      все
      нц для ТекЦифра от 0 до 9
            ЦифраВСудоку[ТекЦифра] := нет
      кц
      НачСтрок := div(НСтроки - 1, 3) * 3 + 1
      НачСтолб := div(НСтолбца - 1, 3) * 3 + 1
      КонСтрок := НачСтрок + 2
      КонСтолб := НачСтолб + 2
      нц для ТекСтрок от 1 до N
            ЦифраВСудоку[Судоку[НСтроки, ТекСтрок]] := да
            ЦифраВСудоку[Судоку[ТекСтрок, НСтолбца]] := да
      кц
      нц для ТекСтрок от НачСтрок до КонСтрок
            нц для ТекСтолб от НачСтолб до КонСтолб
                  ЦифраВСудоку[Судоку[ТекСтрок, ТекСтолб]] := да
            кц
      кц
      Рез := нет
      ТекЦифра := 1
      нц пока не Рез и (ТекЦифра <= 9)
            если не ЦифраВСудоку[ТекЦифра]
                  то
                        СчЗамен := СчЗамен + 1
                        Судоку[НСтроки, НСтолбца] := ТекЦифра
                        если Решить
                              то
                                    Рез := да
                                    выход
                        все
                        Судоку[НСтроки, НСтолбца] := 0
            все
            ТекЦифра := ТекЦифра + 1
      кц
      знач := Рез
кон
```

Листинг 2. Полный текст результатов работы программы

```
+---+---+---+                    +---+---+---+
|000|006|201|                    |845|796|231|
|070|510|060|                    |379|512|468|
|006|080|059|                    |216|384|759|
+---+---+---+                    +---+---+---+
|120|000|000|                    |124|635|897|
|080|940|500|                    |683|947|512|
|590|000|000|                    |597|821|346|
+---+---+---+                    +---+---+---+
|001|050|073|                    |461|258|973|
|030|460|080|                    |732|469|185|
|000|003|604|                    |958|173|624|
+---+---+---+                    +---+---+---+

                                 Число замен: 1287
```

Заключение

Применение описанного в работе подхода позволит решать и многие другие задачи, которые могут быть сведены к рекурсивному перебору с возвратом. Наличие отлаженной программы на школьном алгоритмическом языке упростит его адаптацию для конкретных случаев.

Список литературы

1. *Вирт Н.* Алгоритмы + структуры данных = программы. – Москва: Мир, 1985. – 406 с.

2. Судоку – Википедия [Текст] / Режим доступа: https://ru.wikipedia.org/wiki/%D1%F3%E4%EE%EA%F3

3. *Ткаченко К.С.* Реализация решения головоломки судоку: подход на основе полного перебора // Системный администратор. Вып. 5 (150)/2015. – Москва: ООО «Синдикат 13», 2015. – ISSN: 1813-5579. – С. 85–87.

4. *Ткаченко К.С.* Программная реализация нахождения решения головоломки «Судоку» в 1С // Системный администратор. Вып. 11 (180)/2017. – Москва: ООО «Издательский дом “Положевец и партнеры”», 2017. – ISSN: 1813-5579. – С. 61–63.