Нормальный алгоритм Маркова для решения олимпиадной задачи по информатике «Лягушачьи игры»

Ю. В. Кулаков, email: kulak@list.ru

Тамбовский государственный технический университет

Аннотация. *Разработан алгоритм решения олимпиадной задачи «Лягушачьи игры», основанный на понятии нормального алгоритма Маркова, который доказывает алгоритмическую разрешимость этой задачи для всех допустимых вариантов исходной рассадки лягушек с конечным числом зелёных и коричневых лягушек.*

Ключевые слова: *олимпиадная задача «Лягушачьи игры», алгоритмическая разрешимость задачи, исходная рассадка лягушек, требуемая рассадка лягушек, нормальный алгоритм Маркова, распознаватель вхождения, оператор подстановки.*

Введение

Олимпиадная задача по информатике и программированию «Лягушачьи игры» была предложена организаторами в качестве одной из задач II Всероссийской заочной олимпиады по информатике [1].

Заметим, что данную олимпиадную задачу, вероятно, разработали на основе достаточно известной математической головоломки про шесть лягушек двух различных цветов [2] с расширением до любого конечного числа участвующих в ней зелёных и коричневых лягушек.

Хотя под головоломками обычно понимают непростые задачи, для решения которых, прежде всего, требуется сообразительность, для решения задачи «Лягушачьи игры» необходимо разработать алгоритм, использующий специальные знания высокого уровня.

Упомянутые специальные знания для решения олимпиадной задачи «Лягушачьи игры» нужны, поскольку она, по сути, является перестановочной головоломкой с набором подвижных объектов, перемещаемых с места на место по определённым правилам, которая имеет далеко не очевидный алгоритм решения.

1. Постановка олимпиадной задачи

В тридесятом царстве в новогодние праздники все лягушки собираются на самом большом болоте, чтобы поиграть в замечательную игру. Всего в этом царстве живет N зелёных лягушек и M коричневых. Для игры они выбирают на болоте N + M + 1 кочку, на первые N кочек

слева садятся зелёные лягушки, а на последние M – коричневые (т. е. между ними находится одна кочка, на которой никто не сидит). Зелёные лягушки садятся лицом к коричневым лягушкам, а коричневые – к зелёным. Кочки настолько маленькие, что развернуться на них, не свалившись в болото, совершенно не возможно. Поэтому лягушки могут двигаться только вперед и не могут разворачиваться.

На каждом ходе игры одна из лягушек перепрыгивает с той кочки, где она сидит, на свободную кочку. При этом лягушка может прыгнуть на соседнюю кочку вперед, либо перепрыгнуть через одну кочку, если соседняя занята.

Чтобы праздник удался, зелёные лягушки должны оказаться на последних кочках, а коричневые – на первых. Порядок, в котором лягушки окажутся на кочках, не важен. Так как на праздник каждый раз приходит разное количество лягушек, то им каждый год приходится придумывать очередность прыжков. Напишите программу, которая поможет лягушкам составить план прыжков.

Формат входных данных.

Во входном файле записаны два числа N и M ($1 \le N \le 1000$, $1 \le M \le 1000$) – количество зелёных и коричневых лягушек соответственно.

Формат выходных данных.

Выведите последовательность прыжков лягушек для достижения поставленной цели. Каждый прыжок можно задать одним числом – номером прыгающей лягушки (поскольку свободная кочка всегда ровно одна). Пронумеруем всех лягушек в соответствии с их начальным положением. Зелёные лягушки будут пронумерованы числами от 1 до N, а коричневые – с N + 1 до N + M в порядке слева направо.

Если же достичь требуемой рассадки лягушек нельзя, выведите одно число минус 1.

2. Разработка алгоритма решения задачи

При разработке алгоритма решения задачи «Лягушачьи игры» применим понятие нормального алгоритма Маркова [3].

Для установления закономерностей в процессе прыжков лягушек до достижения их конечной рассадки и построения нормального алгоритма Маркова рассмотрим сначала случай трёх зелёных и трёх коричневых лягушек.

В этом случае исходная рассадка лягушек будет выглядеть так, как показано на рис. 1.

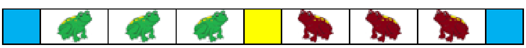


Рис. 1. Исходная рассадка лягушек

Здесь лягушки сидят на шести из семи кочек, жёлтая ячейка таблицы обозначает свободную кочку, синие ячейки – свободную воду болота.

Формализуем для разрабатываемого нормального алгоритма Маркова исходную рассадку лягушек словом «–1110222–» в абстрактном алфавите {0, 1, 2, –}, где символ «0» обозначает свободную кочку; символ «1» – зелёную лягушку; символ «2» – коричневую лягушку; символ «–» – свободную воду болота.

В результате прыжка зелёной лягушки вправо на свободную кочку исходная рассадка лягушек преобразуется в рассадку лягушек после первого прыжка, представленную на рис. 2.

Рис. 2. Рассадка лягушек после первого прыжка

В формализованном представлении в результате прыжка зелёной лягушки слово «–1110222–» преобразуется в слово «–1101222–».

Заметим, что для реализации этого преобразования исходной рассадки лягушек в рассадку после первого прыжка необходимо и достаточно, чтобы первый распознаватель вхождения нормального алгоритма Маркова $РВ_1$ распознал вхождение подслова «1102» в слово «–1110222–» и первый оператор подстановки $ОП_1$ подставил в слово «–1110222–» вместо подслова «1102» подслово «1012». Обозначим подстановку $ОП_1$ нормального алгоритма Маркова как «1102 → 1012».

В результате прыжка коричневой лягушки влево на свободную кочку через зелёную лягушку текущая рассадка лягушек преобразуется в рассадку лягушек после второго прыжка, представленную на рис. 3.

Рис. 3. Рассадка лягушек после второго прыжка

В формализованном представлении в результате прыжка коричневой лягушки через зелёную лягушку слово «–1101222–» преобразуется в слово «–1121022–».

Для реализации этого преобразования в рассадку лягушек после второго прыжка необходимо и достаточно, чтобы второй распознаватель вхождения нормального алгоритма Маркова $РВ_2$ распознал вхождение подслова «1012» в слово «–1101222–» и второй оператор подстановки $ОП_2$ подставил в слово «–1101222–» вместо подслова «1012» подслово

«1210». Обозначим подстановку $ОП_2$ нормального алгоритма Маркова через «1012 → 1210».

В результате прыжка коричневой лягушки влево на свободную кочку текущая рассадка лягушек преобразуется в рассадку лягушек после третьего прыжка, представленную на рис. 4.

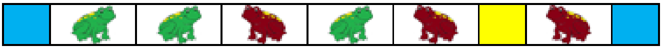


Рис. 4. Рассадка лягушек после третьего прыжка

В формализованном представлении в результате прыжка коричневой лягушки на свободную кочку слово «–1121022–» преобразуется в слово «–1121202–».

Для реализации этого преобразования третий распознаватель вхождения $РВ_3$ должен распознать вхождение подслова «2102» в слово «–1121022–» и третий оператор подстановки $ОП_3$ должен подставить в слово «–1121022–» вместо подслова «2102» подслово «2120». Обозначим подстановку $ОП_3$ нормального алгоритма Маркова через «2102 → 2120».

В результате прыжка зелёной лягушки вправо на свободную кочку через коричневую лягушку рассадка лягушек после третьего прыжка преобразуется в рассадку лягушек после четвёртого прыжка, представленную на рис. 5.

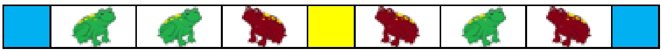


Рис. 5. Рассадка лягушек после четвёртого прыжка

В формализованном представлении в результате данного прыжка зелёной лягушки через коричневую лягушку слово «–1121202–» преобразуется в слово «–1120212–».

Для реализации этого преобразования четвёртый распознаватель вхождения $РВ_4$ должен распознать вхождение подслова «1202» в слово «–1121202–» и четвёртый оператор подстановки $ОП_4$ должен подставить в слово «–1121202–» вместо подслова «1202» подслово «0212». Обозначим подстановку $ОП_4$ нормального алгоритма Маркова через «1202 → 0212».

Проведённые исследования показали, что для реализации остальных прыжков лягушек до достижения их конечной рассадки, причём для любых допустимых по условию олимпиадной задачи количеств зелёных и коричневых лягушек, может потребоваться, кроме описанных выше четырёх операторов подстановки $ОП_1$ – $ОП_4$, ещё одиннадцать операторов: $ОП_5$ «10– → 01–», $ОП_6$ «1201 → 0211», $ОП_7$ «1011 → 0111», $ОП_8$ «–102 → –012», $ОП_9$ «–012 → –210», $ОП_{10}$ «–202 → –220», $ОП_{11}$ «2012 → 2210», $ОП_{12}$ «120– → 021–», $ОП_{13}$ «–021 → –201», $ОП_{14}$ «101– → 011–» и $ОП_{15}$ «2202 → 2220».

В таблице представлены результаты исследования по использованию операторов подстановки при решении задач различной размерности. При этом под задачей размерности (N × M) понимается задача с N зелёными и M коричневыми лягушками. Каждой строке таблицы взаимно однозначно сопоставлен i-ый оператор подстановки $ОП_i$, а каждому столбцу – j-ая размерность задачи. Единица в ячейке (i, j) говорит о том, что i-ый оператор подстановки используется при решении задачи j-ой размерности. В противном случае в ячейке (i, j) таблицы записан 0 (нуль).

Таблица

Оператор подстановки	Размерность задачи								
	1 × 1	1 × 2	1 × 3	2 × 1	2 × 2	2 × 3	3 × 1	3 × 2	3 × 3
$ОП_1$	0	0	0	1	1	1	1	1	1
$ОП_2$	0	0	0	1	1	1	1	1	1
$ОП_3$	0	1	1	0	1	1	0	1	1
$ОП_4$	0	0	1	0	1	1	0	1	1
$ОП_5$	1	0	1	1	0	1	1	0	1
$ОП_6$	0	0	0	1	0	1	1	1	1
$ОП_7$	0	0	0	0	0	0	1	0	1
$ОП_8$	1	1	1	0	0	0	1	1	1
$ОП_9$	1	1	1	0	0	0	1	1	1
$ОП_{10}$	0	1	1	0	0	0	0	1	1
$ОП_{11}$	0	0	1	0	1	1	0	0	1
$ОП_{12}$	0	1	0	0	1	0	0	1	0
$ОП_{13}$	0	0	0	1	1	1	0	0	0
$ОП_{14}$	0	0	0	0	1	0	0	1	0
$ОП_{15}$	0	0	0	0	0	1	0	0	0

3. Граф-схема сконструированного нормального алгоритма Маркова

Сконструирован нормальный алгоритм Маркова для решения олимпиадной задачи по информатике «Лягушачьи игры» при всех допустимых вариантах исходной рассадки лягушек с конечным числом зелёных и коричневых лягушек, граф-схема которого представлена на рис. 6.

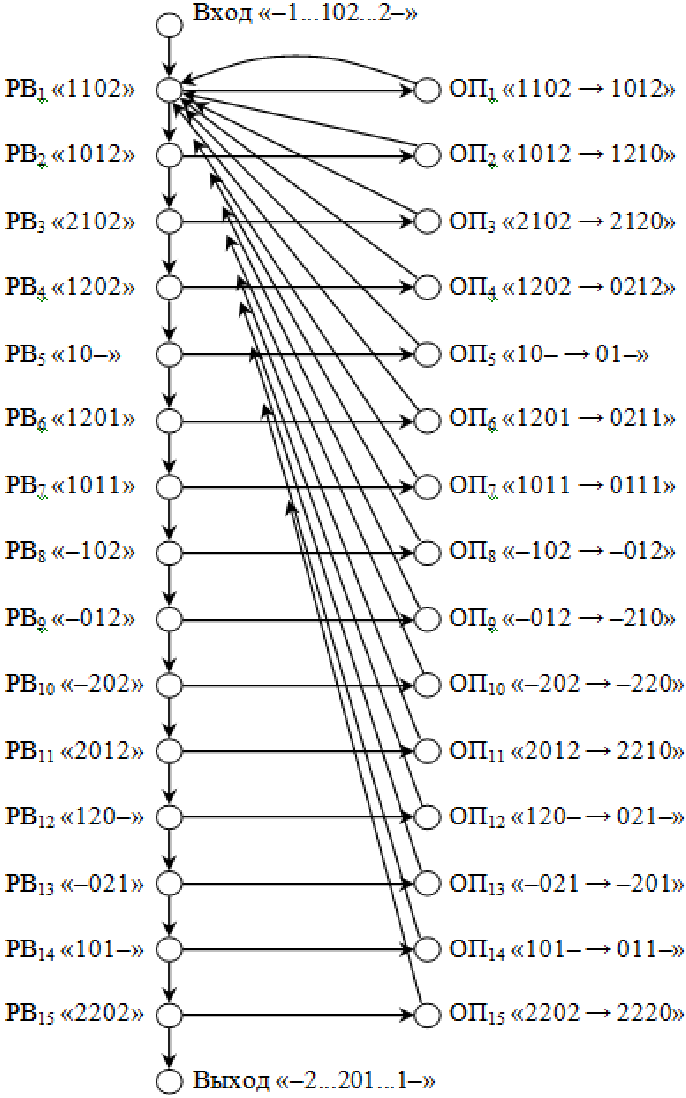
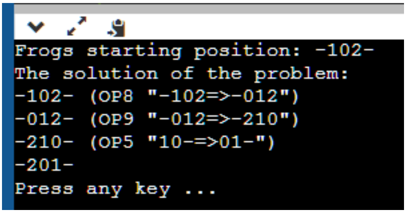


Рис. 6. Граф-схема сконструированного нормального алгоритма Маркова

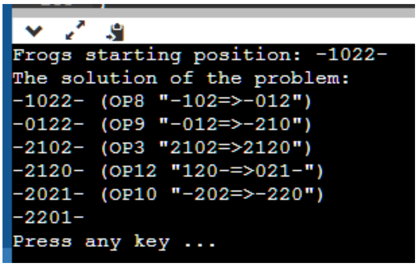
4. Программная реализация сконструированного нормального алгоритма Маркова

Сконструированный нормальный алгоритм Маркова для решения олимпиадной задачи по информатике «Лягушачьи игры» реализован в виде программы на алгоритмическом языке Си. Скриншоты результатов выполнения этой программы в онлайн компиляторе, доступном в Интернете по адресу https://www.onlinegdb.com/online_c++_compiler, для каждой из девяти различных размерностей задачи из приведённой выше таблицы представлены на рис. 7–15.



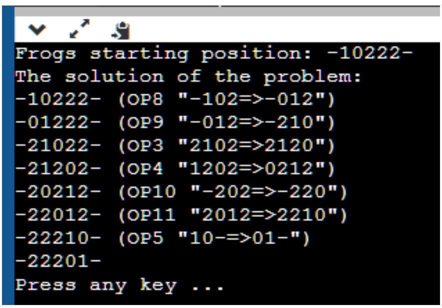
```
Frogs starting position: -102-
The solution of the problem:
-102- (OP8 "-102=>-012")
-012- (OP9 "-012=>-210")
-210- (OP5 "10-=>01-")
-201-
Press any key ...
```

Рис. 7. Скриншот решения задачи размерности 1 × 1



```
Frogs starting position: -1022-
The solution of the problem:
-1022- (OP8 "-102=>-012")
-0122- (OP9 "-012=>-210")
-2102- (OP3 "2102=>2120")
-2120- (OP12 "120-=>021-")
-2021- (OP10 "-202=>-220")
-2201-
Press any key ...
```

Рис. 8. Скриншот решения задачи размерности 1 × 2



```
Frogs starting position: -10222-
The solution of the problem:
-10222- (OP8 "-102=>-012")
-01222- (OP9 "-012=>-210")
-21022- (OP3 "2102=>2120")
-21202- (OP4 "1202=>0212")
-20212- (OP10 "-202=>-220")
-22012- (OP11 "2012=>2210")
-22210- (OP5 "10-=>01-")
-22201-
Press any key ...
```

Рис. 9. Скриншот решения задачи размерности 1 × 3

```
Frogs starting position: -1102-
The solution of the problem:
-1102- (OP1 "1102=>1012")
-1012- (OP2 "1012=>1210")
-1210- (OP5 "10-=>01-")
-1201- (OP6 "1201=>0211")
-0211- (OP13 "-021=>-201")
-2011-
Press any key ...
```

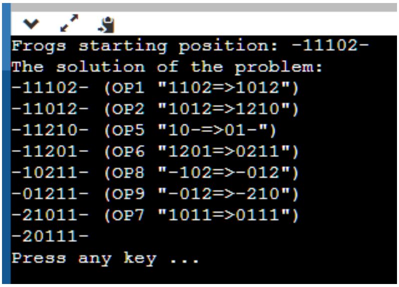
Рис. 10. Скриншот решения задачи размерности 2 × 1

```
Frogs starting position: -11022-
The solution of the problem:
-11022- (OP1 "1102=>1012")
-10122- (OP2 "1012=>1210")
-12102- (OP3 "2102=>2120")
-12120- (OP12 "120-=>021-")
-12021- (OP4 "1202=>0212")
-02121- (OP13 "-021=>-201")
-20121- (OP11 "2012=>2210")
-22101- (OP14 "101-=>011-")
-22011-
Press any key ...
```

Рис. 11. Скриншот решения задачи размерности 2 × 2

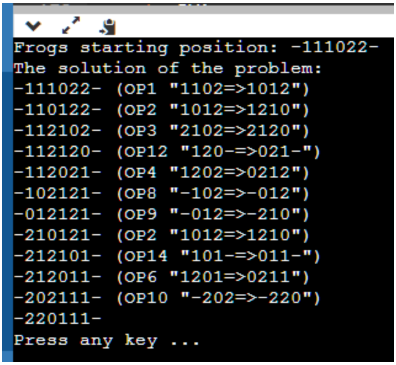
```
Frogs starting position: -110222-
The solution of the problem:
-110222- (OP1 "1102=>1012")
-101222- (OP2 "1012=>1210")
-121022- (OP3 "2102=>2120")
-121202- (OP4 "1202=>0212")
-120212- (OP4 "1202=>0212")
-021212- (OP13 "-021=>-201")
-201212- (OP11 "2012=>2210")
-221012- (OP2 "1012=>1210")
-221210- (OP5 "10-=>01-")
-221201- (OP6 "1201=>0211")
-220211- (OP15 "2202=>2220")
-222011-
Press any key ...
```

Рис. 12. Скриншот решения задачи размерности 2 × 3



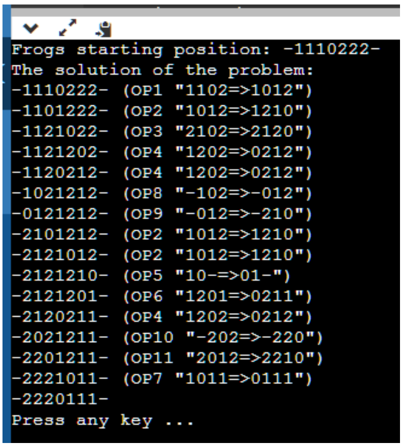
```
Frogs starting position: -11102-
The solution of the problem:
-11102- (OP1 "1102=>1012")
-11012- (OP2 "1012=>1210")
-11210- (OP5 "10-=>01-")
-11201- (OP6 "1201=>0211")
-10211- (OP8 "-102=>-012")
-01211- (OP9 "-012=>-210")
-21011- (OP7 "1011=>0111")
-20111-
Press any key ...
```

Рис. 13. Скриншот решения задачи размерности 3 × 1



```
Frogs starting position: -111022-
The solution of the problem:
-111022- (OP1 "1102=>1012")
-110122- (OP2 "1012=>1210")
-112102- (OP3 "2102=>2120")
-112120- (OP12 "120-=>021-")
-112021- (OP4 "1202=>0212")
-102121- (OP8 "-102=>-012")
-012121- (OP9 "-012=>-210")
-210121- (OP2 "1012=>1210")
-212101- (OP14 "101-=>011-")
-212011- (OP6 "1201=>0211")
-202111- (OP10 "-202=>-220")
-220111-
Press any key ...
```

Рис. 14. Скриншот решения задачи размерности 3 × 2



```
Frogs starting position: -1110222-
The solution of the problem:
-1110222- (OP1 "1102=>1012")
-1101222- (OP2 "1012=>1210")
-1121022- (OP3 "2102=>2120")
-1121202- (OP4 "1202=>0212")
-1120212- (OP4 "1202=>0212")
-1021212- (OP8 "-102=>-012")
-0121212- (OP9 "-012=>-210")
-2101212- (OP2 "1012=>1210")
-2121012- (OP2 "1012=>1210")
-2121210- (OP5 "10-=>01-")
-2121201- (OP6 "1201=>0211")
-2120211- (OP4 "1202=>0212")
-2021211- (OP10 "-202=>-220")
-2201211- (OP11 "2012=>2210")
-2221011- (OP7 "1011=>0111")
-2220111-
Press any key ...
```

Рис. 15. Скриншот решения задачи размерности 3 × 3

Приведённые скриншоты работы программы подтверждают данные представленной выше таблицы об использовании операторов подстановки $ОП_1$ – $ОП_{15}$ при решении задачи «Лягушачьи игры» для всевозможных исходных данных при размерности задачи от 1 × 1 до 3 × 3 включительно.

Заметим, что по разработанному алгоритму успешно решается задача «Лягушачьи игры» размерностью выше, чем 3 × 3, а при этом используются операторы подстановки из того же множества операторов подстановки $\{ОП_1, \ldots, ОП_{15}\}$. Например, при решении задачи размерности 10 × 10 (с десятью зелёными и десятью коричневыми лягушками) используются операторы подстановки $ОП_1$ – $ОП_7$ и $ОП_{11}$ – $ОП_{15}$, при решении задачи размерности 15 × 20 (с пятнадцатью зелёными и двадцатью коричневыми лягушками) используются операторы подстановки $ОП_1$ – $ОП_4$, $ОП_6$ – $ОП_{12}$, $ОП_{14}$ и $ОП_{15}$, а при решении задачи размерности 20 × 15 (с двадцатью зелёными и пятнадцатью коричневыми лягушками) используются операторы подстановки $ОП_1$ – $ОП_7$, $ОП_{11}$, $ОП_{13}$ и $ОП_{15}$.

Заключение

Разработан алгоритм решения олимпиадной задачи «Лягушачьи игры», представляющий собой нормальный алгоритм Маркова и поэтому доказывающий алгоритмическую разрешимость этой задачи, под которой понимается достижение требуемой конечной рассадки лягушек для всех допустимых вариантов исходной рассадки лягушек с конечным числом зелёных и коричневых лягушек.

Данная задача и представленный алгоритм её решения могут быть использованы учащимися при подготовке к участию в олимпиадах по программированию, при изучении темы «Нормальные алгоритмы Маркова» и при подготовке к сдаче Единого государственного экзамена по информатике.

Список литературы

1. Олимпиады по программированию. II Всероссийская заочная олимпиада школьников по информатике [Электронный ресурс]: Задача K. Лягушачьи игры. – Режим доступа: https://www.olympiads.ru/zaoch/2007/problems/k.shtml
2. Блог вебмастера [Электронный ресурс]: Шесть лягушек поменять местами. – Режим доступа: https://teh-fed.ru/6-ljagushek-pomenjat-mestami/
3. Акулов О.А. Информатика: базовый курс / О.А. Акулов, Н.В. Медведев. – М.: Омега-Л, 2007. – 560 с.