

Майер Р.В.
РЕШЕНИЕ УЧЕБНОЙ ЗАДАЧИ КАК ПОИСК
ВЫХОДА ИЗ ВИРТУАЛЬНОГО ЛАБИРИНТА

Учебной задачей в самом широком смысле называется любое задание, которое получает учащийся от учителя с целью обучения. Несколько сузим это понятие, исключив творческие задания (написание сочинения или рисование картины), так как творческая деятельность плохо формализуется, и эти задания могут быть выполнены огромным числом различных способов. В результате останутся задачи, решение которых требует последовательного выполнения ограниченного числа операций в определенном порядке (решить уравнение, нарисовать график, заполнить таблицу, собрать электрическую цепь).

Чтобы решить задачу школьнику следует: 1) правильно выбрать алгоритм решения (он может быть не самым оптимальным, но все равно должен приводить к результату); 2) правильно выполнить все операции. К **задачам первого типа** будем относить те задачи, для решения которых учащийся использует алгоритмический подход. Например, нахождение корней квадратного уравнения школьником, который знает все необходимые формулы и последовательность действий (алгоритм решения). К **задачам второго типа** отнесем те задачи, алгоритм решения которых неизвестен, и человек вынужден применять метод перебора, эвристический метод, интуитивные рассуждения. Часто учебная задача для данного школьника частично является задачей первого типа, а частично – второго типа.

При самостоятельном решении задач основную роль играет замкнутая цепь управления, реализуемая в сознании учащегося. Учащемуся сообщают условия задачи, исходные данные и дают задание, что необходимо найти. При этом возможны два варианта: 1. Учащийся может убедиться в правильности своего ответа. Например, он решает уравнение и, найдя его корни, может путем подстановки убедиться в правильности решения. 2. Учащийся не может проверить в правильность своего решения, ему не с чем сравнить полученный результат.

Решение многих математических, физических и других задач сводится к последовательности операций, выполняемых в определенном порядке. Часто заранее неизвестно, какие именно операции требуются и в какой последовательности их следует выполнять. Как правило, испытуемый способен осуществить значительно большее количество операций, чем это необходимо для достижения цели. Решение задачи аналогично поиску выхода из виртуального лабиринта, соединяющего пункт А с пунктом В и представимого в виде графа. Лабиринт имеет прямой путь, соответствующий правильному решению задачи, а также множество различных ответвлений, заканчивающихся тупиками. Если задача имеет несколько решений, то ей соответствует лабиринт с несколькими различными путями из А в В. Самый короткий путь отвечает оптимальному способу решения задачи.

Всем известны классические опыты с мышью, бегающей по лабиринту и ищущий путь к еде. Во втором и третьем испытании мышь быстрее достигает цели, так как она “запомнила решение задачи”. Нечто похожее происходит и с чело-

веком: решая аналогичную задачу второй, третий или четвертый раз, он быстрее “находит выход из виртуального лабиринта”.

Автором была создана компьютерная программа, моделирующая лабиринт, изображенный на рис. 1. При ее запуске испытуемому сообщается, что он находится в комнате Z (начало), которая имеет три двери с буквами А, В и С. За дверью А находится комната А, за дверью В — комната В, за дверью С — комната С. Испытуемый случайно выбирает одну из дверей и по воображаемому коридору проходит в следующую комнату. Если он открыл дверь С, то он попадает в комнату С, где ему предлагается выбрать двери Z, Н или N. После выбора той или иной двери, испытуемый попадает в соответствующую комнату, где он делает следующий выбор. Когда человек попадает в комнату Р, на экране появляется сообщение о том, что он вышел из лабиринта (дошел до его конца), то есть справился с задачей. Программа ПР-1 общается с испытуемым только в текстовом режиме, выбор той или иной двери осуществляется с помощью клавиатуры и регистрируется в текстовом файле data.txt. В этом же файле записывается время, требуемое для выбора той или двери на каждом этапе. Если незначительно изменить программу, то можно сделать так, чтобы испытуемый на любом этапе имел возможность, нажав на соответствующую клавишу, вернуться к началу пути Z и предпринять еще одну попытку выхода из лабиринта.

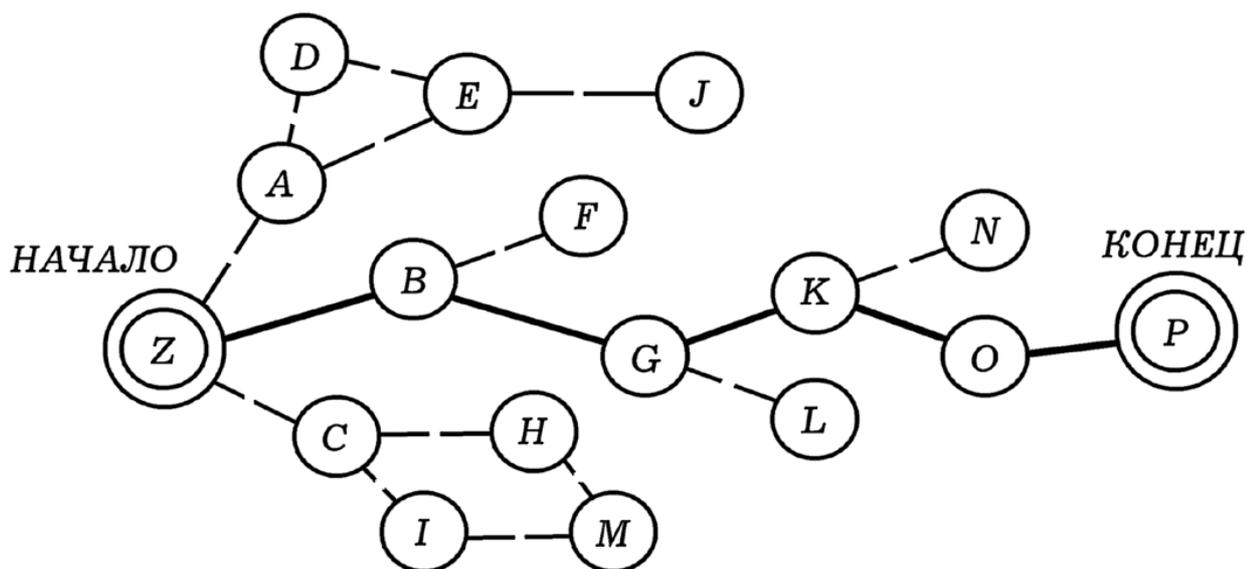


Рис. 1. Виртуальный лабиринт, моделируемый компьютерной программой.

Данная программа была апробирована со несколькими школьниками. Перед ними была поставлена задача, пройти лабиринт за минимальное время. Результаты расшифровки пути решения этой задачи дали следующее. При первом прохождении лабиринта учащиеся ведут себя как вероятностные автоматы случайным образом выбирая ту или иную дверь. Попав в тупик J, они возвращаются обратно в Z, кружат в цикле С–Н–М–I, затем случайно находят правильное решение ZBГKOP. При повторном прохождении того же самого лабиринта на решение задачи затрачивается существенно меньше времени: испытуемый сразу выбирает

комнату В и находит путь к Р, ошибаясь гораздо реже. При третьем прохождении лабиринта испытуемый безошибочно выбирает правильный путь ZBGKOP.

Затем была поставлена более сложная задача: полностью исследовать лабиринт и на листочке бумаги нарисовать граф, изображающий комнаты, соединенные коридорами (то есть получить рис. 1). Как показывает практика, справиться с этим заданием сложнее. Испытуемые делают ошибки, не указывая все соединения (коридоры) между комнатами, либо рисуя на своем листочке две разные комнаты с одинаковыми буквами.

```

A 1.2634900136E+01
AD 6.5024999534E+00
ADE 7.4599999292E+00
ADEA 6.9775999414E+00
ADEAZ 6.9919999410E+00
ADEAZC 6.5076999533E+00
ADEAZCI 6.3390999575E+00
ADEAZCIM 6.8378999449E+00
ADEAZCIMH 6.9929999410E+00
ADEAZCIMHC 6.9891999411E+00
ADEAZCIMHCI 6.2592999596E+00
ADEAZCIMHCIM 7.3922999309E+00
ADEAZCIMHCIMH 7.5107999279E+00
ADEAZCIMHCIMHM 7.0250999402E+00
ADEAZCIMHCIMHMC 6.5862999513E+00
ADEAZCIMHCIMHMCI 7.5237999276E+00
ADEAZCIMHCIMHMCIC 6.8854999437E+00
ADEAZCIMHCIMHMCICZ 6.7917999461E+00
ADEAZCIMHCIMHMCICZB 7.0314999400E+00
ADEAZCIMHCIMHMCICZBG 7.0286999401E+00
ADEAZCIMHCIMHMCICZBGK 7.0246999402E+00
ADEAZCIMHCIMHMCICZBGKO 7.0388999399E+00
ADEAZCIMHCIMHMCICZBGKOP 7.0452999397E+00

```

Рис. 2. Пример прохождения виртуального лабиринта (файл data.txt).

Возникает вопрос о длине решения задачи о количестве действий, которые следует предпринять, чтобы выйти из лабиринта. Если испытуемому сообщили правильное решение ZBGKOP, то он должен выполнить 5 шагов, то есть $L_{\min} = 5$. Понятно, что длина пути зависит от выбора, совершаемого учащимся, который носит случайный характер. Находясь в комнате Z, человек с вероятностью $1/3$ выбирает один из трех путей решения задачи (двери А, В или С). Двигаясь по пути А, он с равными вероятностями $1/2$ совершает 3 или 4 шага до тупика J, а затем возвращается обратно. Поэтому длина пути от Z до J и обратно $L_1 = 7$. Аналогично, длина пути от Z до М и обратно $L_3 = 6$. Найдем длину пути от Z до G. От Z до В – 1 шаг, далее человек с вероятностью $1/2$ идет по пути В–F–В–G, совершая 3 шага, либо с той же вероятностью сразу по пути В–G – совершая 1 шаг. Получаем, что длина $L_{ZG} = 1 + 3 \cdot 0,5 + 1 \cdot 0,5 = 3$. Рассуждая аналогично, получаем, что $L_{GK} = 2$, $L_{KO} = 2$, $L_{OP} = 1$, поэтому средняя длина пути по ветви В равна: $L_2 = 3 + 2 + 2 + 1 = 8$. В результате для выхода из лабиринта испы-

туемый, действуя совершенно случайно, должен совершить в среднем $L_1/3 + L_2/3 + L_3/3 = 7/3 + 8/3 + 6/3 = 7$ шагов.

При решении теоретических задач, учащийся, почувствовав, что движется в неправильном направлении или попал в тупик, в любой момент может вернуться к началу пути и все повторить снова. Обычно школьник понимает, что данная задача требует в 2 – 3 – 4 действия. Поэтому, если учащийся выполнил несколько достаточно сложных шагов и не пришел к результату, то значит он находится на неправильном пути. Школьник снова возвращается к началу и предпринимает повторную попытку решить задачу.

На основе программы ПР–1, работающей в текстовом режиме, возможно написать программу, создающую графическую модель движения человека по виртуальному лабиринту. При ее запуске на экране ПЭВМ появляются изображения комнаты и человека, стоящего перед несколькими дверями с различными буквами, он открывает дверь, идет по коридору, попадает в другую комнату и т.д. Такая программа должна оценивать быстроту прохождения лабиринта и позволять изменять его карту. Испытуемый должен не только пройти весь лабиринт, но нарисовать его схему. Эта программа может быть использована для развития детей младшего и среднего школьного возраста.

```
uses dos, crt;                                     { ПР - 1 }
var S,aa: string; t: real; ff: text;               { Free Pascal }
Label Z,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P;
BEGIN clrscr; Assign(FF,'data.txt'); Rewrite(FF);
Z: Writeln('*** NACHALO. PUNKT Z *** ');
Writeln('VIBERI ODNU DVER: A, B, C ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='A' then goto A; If aa='B' then goto B;
If aa='C' then goto C; Writeln('OSHIBKA'); goto Z;
A: Writeln('VIBERI ODNU DVER: D, E, Z ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='Z' then goto Z; If aa='D' then goto D;
If aa='E' then goto E; Writeln('OSHIBKA'); goto A;
B: Writeln('VIBERI ODNU DVER: F, G, Z ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='Z' then goto Z; If aa='F' then goto F;
If aa='G' then goto G; Writeln('NEVERNO'); goto B;
C: Writeln('VIBERI ODNU DVER: H, I, Z ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='Z' then goto Z; If aa='H' then goto H;
If aa='I' then goto I; Writeln('NEVERNO'); goto C;
```

```
D: Writeln('VIBERI ODNU DVER: A, E ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='A' then goto A; If aa='E' then goto E;
Writeln('NEVERNO'); goto D;
E: Writeln('VIBERI ODNU DVER: A, D, J');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='A' then goto A; If aa='D' then goto D;
If aa='J' then goto J; Writeln('NEVERNO'); goto E;
F: Writeln('TUPIK. VIBERI DVER: B ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='B' then goto B; Writeln('NEVERNO'); goto F;
G: Writeln('VIBERI ODNU DVER: B, L, K ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='B' then goto B; If aa='L' then goto L;
If aa='K' then goto K; Writeln('NEVERNO'); goto G;
H: Writeln('VIBERI ODNU DVER: C, M ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='C' then goto C; If aa='M' then goto M;
Writeln('NEVERNO'); goto H;
I: Writeln('VIBERI ODNU DVER: C, M ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='C' then goto C; If aa='M' then goto M;
Writeln('NEVERNO'); goto I;
J: Writeln('TUPIK. DVER E ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='E' then goto E; Writeln('NEVERNO'); goto J;
K: Writeln('VIBERI ODNU DVER: G, N, O ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='G' then goto G; If aa='N' then goto N;
If aa='O' then goto O; Writeln('NEVERNO'); goto K;
L: Writeln('TUPIK. DVER G ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='G' then goto G; Writeln('NEVERNO'); goto L;
M: Writeln('VIBERI ODNU DVER: H, I ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
```

```
If aa='H' then goto H; If aa='I' then goto I;
Writeln('NEVERNO'); goto M;
N: Writeln('TUPIK. DVER K ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='K' then goto K; Writeln('NEVERNO'); goto N;
O: Writeln('VIBERI ODNU DVER: K, P ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='K' then goto K; If aa='P' then goto P;
Writeln('NEVERNO'); goto O;
P: Writeln('** KONEC PUTI ** ');
Repeat until Keypressed; Close(FF);
END.
```

Литература

1. Адамар, Ж. Исследование психологии процесса изобретения в области математики [Текст] / Ж. Адамар. — М.: Сов. радио, 1970. — 152 с.
2. Егидес, А. П., Егидес Е. М. Лабиринты мышления или учеными не рождаются [Текст] / А. П. Егидес, Е. М. Егидес. — М.: АСТ-ПРЕСС КНИГА, 2004. — 320 с.
3. Шапиро, С. И. От алгоритмов — к суждениям (эксперименты по обучению элементам математического мышления) [Текст] / С.И. Шапиро. — М.: Сов. радио, 1973. — 288 с.