

П.В.Гляков, кандидат физико-математических наук, доцент

ИГРОВАЯ ФОРМА ОБУЧЕНИЯ АЛГОРИТМИЗАЦИИ КУЛЬТУРОЛОГОВ-МЕНЕДЖЕРОВ

Учебный план подготовки культурологов-менеджеров включает дисциплину “Алгоритмизация и программирование”. Наиболее трудной частью этой дисциплины является алгоритмизация, которая содержит элементы теории сложности алгоритмов. Студенты должны научиться не только разрабатывать алгоритмы для решения задач, но и научиться оценивать их сложность. В качестве вычислительной модели для оценки сложности алгоритмов используется алгоритмический язык [1].

После того как на лекции рассмотрены такие вопросы, как размер задачи, временная сложность алгоритма, емкостная сложность алгоритма, единицы измерения времени и памяти алгоритма, на практических занятиях студенты приступают к вычислению характеристик времени и памяти алгоритмов в худшем случае, т.е. в таком случае, когда алгоритм будет требовать при выполнении наибольшего количества времени.

При рассмотрении этой темы на практическом занятии оказалось целесообразным использовать игровую форму обучения. Начинается она с разделения группы на 3–4 команды. Количество команд зависит от количества студентов в группе и возможности аудитории, в которой проводится занятие. Студенты одной команды должны иметь возможность общаться между собой, не мешая при этом студентам других команд.

В каждой команде должен быть капитан. Он может назначаться преподавателем или избираться студентами группы. В первом случае разделение группы на команды осуществляется капитанами. Происходит это так. Каждый капитан поочередно набирает в свою команду по одному студенту. Так происходит до тех пор, пока все студенты не будут распределены. Такой подход позволяет по возможности учесть пожелания капитанов. Составляющими элементами этих пожеланий являются: уровень знания алгоритмизации, уровень коммуникабельности и личностная совместимость.

Во втором случае студенты каждой команды сами выбирают себе капитана. При этом они руководствуются в основном теми же критериями, как и в первом случае.

Умение студентов работать в командах является важным для будущих культурологов-менеджеров, поскольку в своей профессиональной деятельности многие из них будут руководителями (менеджерами) разных уровней. Иногда психологически важно при работе в командах напомнить студентам афоризм: “Никто из Вас не знает так много, как все Вы вместе”.

Далее преподаватель формулирует условие задачи, алгоритм решения которой надо построить и оценить его эффективность. Задача должна быть такая, чтобы допускала достаточную вариативность решения. Удобно для этой цели взять задачу последовательного поиска.

Условие. Дан массив целых чисел a размерности n . Требуется определить, есть ли в нем элемент a_i , равный некоторому целому числу b . Если такой элемент есть, то строковой переменной c присвоить значение “да”, в противном случае присвоить значение “нет”.

Когда команды разработают свои алгоритмы, капитаны команд выходят к доске и записывают алгоритмы решения задачи. Доска для этих целей должна быть стандартная с разворотами, чтобы студенты могли видеть решения всех команд. Иногда в представленных решениях могут быть ошибки, и студенты команды не видят их или не могут устранить. Тогда преподаватель подбирает такие исходные данные для исполнения алгоритма, на которых алгоритм не даст требуемого решения. Капитану и его команде предлагают исполнить алгоритм и устранить ошибку. Если команда не в состоянии устранить ошибку, преподаватель обращается к другим командам, чтобы они указали, как можно устранить ошибку, или сам предлагает собственный вариант устранения ошибки.

Иногда при решении задачи последовательного поиска капитаны представляют одинаковые алгоритмы. В этом случае больше баллов получит та команда, которая представила свой алгоритм первой. Обычно алгоритмы, представляемые командами на алгоритмическом языке, содержат незначительные отличия, но эти отличия влекут за собой существенные отличия в оценке их эффективности. Обозначим представленные алгоритмы как Поиск 1, Поиск 2, Поиск 3 и Поиск 4.

Идея алгоритма Поиск 1 заключается в следующем. Последовательно просматриваются элементы массива **a** и сравниваются с величиной **b**. Как только будет обнаружен элемент, равный **b**, цикл завершает свою работу, поскольку переменной **i**, используемой для организации цикла, присваивается значение, которое делает условие цикла ложным.

В алгоритме Поиск 2 условие цикла является составным. В этом условии проверяется не только переменная **i**, которая используется для последовательного просмотра элементов массива **a**, но и переменная **c**, которой предварительно установлено значение “нет”. Как только будет найден искомый элемент, этой переменной будет присвоено значение “да” и цикл завершит свою работу.

Алгоритм Поиск 3 не завершает свою работу сразу после нахождения искомого элемента. Он будет осуществлять просмотр всех элементов массива **a**, и каждый раз, когда будет встречаться элемент **a_i**, равный **b**, переменной **c** будет присваиваться значение “да”. Становится понятным, что в среднем случае этот алгоритм будет работать дольше, чем алгоритмы Поиск 1 и Поиск 4.

Цикл в алгоритме Поиск 4 завершает свою работу сразу же после обнаружения искомого элемента массива **a**. Для этой цели так же, как и в алгоритме Поиск 1, переменной **i**, используемой в условии цикла, присваивается значение, которое делает условие цикла ложным. Отличие от алгоритма Поиск 1 состоит в том, что значение переменной **c**, в которой формируется результат выполнения алгоритма, присваивается после завершения работы цикла.

После представления алгоритмов команды должны вычислить характеристики времени $T(n)$ и памяти $M(n)$ для худшего случая. Такой случай соответствует ситуации, в которой массив **a** не содержит элемента **a_i**, равного числу **b**.

Характеристики времени в худшем случае для представленных алгоритмов будут следующие: $T_1(n)=3n+3$, $T_2(n)=4n+4$, $T_3(n)=3n+3$, $T_4(n)=3n+4$. Для всех четырех алгоритмов память, требуемая для их выполнения в худшем случае, будет одинаковой: $M(n)=n+4$. В эту величину входит память, требуемая для размещения исходных данных, результата и вспомогательной переменной **i**.

Из вычисленных характеристик времени выполнения алгоритма в худшем случае видно, что при достаточно большом размере задачи **n** алгоритм Поиск 2 будет выполняться дольше остальных в $4/3$ раза.

Как правило, командам не удастся найти более быстрый алгоритм для решения задачи последовательного поиска, чем рассмотренные выше. Поэтому преподаватель сам знакомит студентов с идеей алгоритма быстрого последовательного поиска (БПП) и сам записывает его на доске [2].

Командам предлагается вычислить для этого алгоритма характеристики времени и памяти в худшем случае. Они будут равны соответственно следующим величинам: $T_{\text{БПП}}(n)=2n+5$, $M_{\text{БПП}}(n)=n+5$.

Сравнение характеристики времени алгоритма БПП с характеристикой времени алгоритма Поиск 1 (который является одним из наиболее быстрых алгоритмов, предложенных командами) при достаточно большом значении n показывает, что алгоритм БПП выполняется в 1,5 раза быстрее, чем алгоритм Поиск 1, говоря другими словами, его использование вместо алгоритма Поиск 1 позволяет экономить электроэнергию в 1,5 раза.

Как и всякая игра, игровая форма проведения занятия имеет дух состязательности. И чтобы этот дух не угасал, необходимо постоянно фиксировать достижения команд. Поэтому на доске ведется учет баллов, заработанных командами. При этом на занятии выделяют три этапа.

На первом этапе команда, разработавшая первой правильный алгоритм решения задачи последовательного поиска, получает 4 балла. Команда, которая была второй, получает 3 балла и т.д. Команда, допустившая ошибку при разработке алгоритма, баллов не получает. Команда, первая обнаружившая ошибку в алгоритме другой команды, получает дополнительно 1 балл.

Второй этап состязания начинается после того, как все четыре алгоритма решения задачи последовательного поиска записаны на доске и в них ликвидированы ошибки. На этом этапе командам предлагается вычислить характеристики времени и памяти разработанных ими алгоритмов. Опять, как и на первом этапе состязания, команда, которая первая вычислила характеристику времени своего алгоритма (и она не содержит ошибок), получает 4 балла. Остальные команды в порядке очередности получают на 1 балл меньше относительно друг друга. Поощрительный балл добавляется той команде, у которой характеристика времени выполнения алгоритма оказалась лучше. За правильное вычисление

характеристики памяти разработанных алгоритмов командам начисляется по одному баллу.

Третий этап начинается после того, как преподаватель напишет на доске алгоритм быстрого последовательного поиска. На этом этапе разыгрываются 2 балла. Из них один начисляется команде, первой вычислившей правильно характеристику времени алгоритма, а второй – команде за вычисление соответственно характеристики памяти алгоритма. Заработанные командами баллы учитываются преподавателем при сдаче экзамена по дисциплине “Алгоритмизация и программирование”.

В результате апробации описанной формы проведения занятия выявлены следующие особенности: высокая мотивация студентов, дух состязательности, интенсивность учебного процесса, эмоциональная окраска занятия, большой объем выполненной учебной работы. Перечисленные особенности свидетельствуют о результативности учебного процесса с использованием игровой формы обучения.

1. Ахо, А. Построение и анализ вычислительных алгоритмов / А.Ахо, Дж.Хопкрофт, Дж.Ульман. – М.: Мир, 1979. – 536 с.

2. Кнут, Д. Искусство программирования для ЭВМ / Д.Кнут. – М.: Мир, 1978. – Т. 3: Сортировка и поиск. – 848 с.