

необходимо учитывать вызовы и риски, связанные с этим процессом, чтобы принимать информированные решения и достичь сбалансированного подхода к использованию ИИ в образовании.

Литература:

1. Norvig, P., & Russell, S. (2010). Artificial Intelligence: A Modern Approach. Prentice Hall.
2. Интернет – ресурс: <https://www.unesco.org/ru/articles/kak-iskusstvennyy-intellekt-mozhet-uluchshit-obrazovanie>

КОМБИНАТОРНЫЕ КОНФИГУРАЦИИ В РЕШЕНИИ ЗАДАЧ В СФЕРЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Лазарев С.Е.,

*обучающийся 1 курса АНО ПО «Колледж предпринимательских
и цифровых технологий», г. Кемерово, Российская Федерация
Научный руководитель – Шабарчина В.Ю., преподаватель*

Математика и программирование – две неразрывно связанные науки. Математика является основой программирования, играя важнейшую роль в различных его областях.

Во-первых, математика – это основа для решения различных задач в области машинного обучения и искусственного интеллекта. Во-вторых, математика играет важнейшую роль в разработке компьютерных игр: программирование трехмерной графики, анимации движения невозможно без знаний в области геометрии и тригонометрии. В-третьих, математика используется при работе с алгоритмами и массивами данных.

Одним из фундаментальных понятий в программировании является понятие «алгоритм». Алгоритм – это последовательность команд управления каким-либо исполнителем [2, с. 10]. В свою очередь алгоритмизация – это процесс разработки алгоритма. Задача написания алгоритма появляется в различных сферах программирования. Для разработки алгоритмов программисту необходимы знания математической логики и комбинаторики.

При работе с массивами данных программисту также необходимы знания в различных областях математики, в частности, комбинаторики. Комбинаторика помогает оптимизировать процессы обработки и хранения данных.

Комбинаторика – область математики, в которой изучается вопрос о том, сколько различных конфигураций, удовлетворяющих тем или иным условиям, можно составить из заданных объектов [1, с. 244].

Комбинаторика в программировании выполняет многие задачи, в том числе, помогает анализировать алгоритмы и тестировать различные программы. Комбинаторика помогает программисту ответить на вопрос «Сколько комбинаций можно составить из данных элементов по конкретным правилам?».

Рассмотрим пример часто встречающейся задачи в программировании. Пусть есть одномерный массив, состоящий из n элементов. Необходимо определить, сколько комбинаций можно составить из элементов, и перебрать все возможные комбинации.

Способ первый. Перестановки – комбинации данного массива, получаемые перестановкой элементов. Количество перестановок находится по формуле:

$$P_n = n! \quad (1)$$

P_n – количество перестановок;

n – количество элементов одномерного массива.

Способ второй. Сочетания представляют собой комбинации определенной длины k , составленные из одномерного массива, состоящего из n элементов. Количество сочетаний находится по формуле:

$$C_n^k = \frac{n!}{(n-k)!k!} \quad (2)$$

C_n^k – количество сочетаний;
 n – количество элементов одномерного массива;
 k – количество элементов необходимой комбинации.

Способ третий. Размещения по сути, являются сочетаниями, в которых важен порядок элементов. Другими словами, это перестановки сочетаний. Количество размещений находится по формуле:

$$A_n^k = \frac{n!}{(n-k)!} \quad (3)$$

A_n^k – количество размещений;
 n – количество элементов одномерного массива;
 k – количество элементов необходимой комбинации.

Способ четвертый. Размещения с повторениями – это все варианты массивов длиной k , на каждой позиции которых может быть любой элемент из множества одномерного массива n . Количество размещений с повторениями находится по формуле:

$$\overline{A_n^k} = n^k \quad (4)$$

$\overline{A_n^k}$ – количество размещений с повторениями;
 n – количество элементов одномерного массива;
 k – количество элементов необходимой комбинации.

Перебор различных комбинаторных конфигураций объектов и выбор среди них наилучшего – довольно часто встречаемая задача в работе программиста. Потому знание алгоритмов генерации наиболее распространенных комбинаторных конфигураций является необходимым условием успешной работы IT-специалиста. Важно также знать количество различных вариантов для каждого типа комбинаторных конфигураций, так как это позволяет реально оценить вычислительную трудоемкость выбранного алгоритма решения той или иной задачи на перебор вариантов и приемлемость этого алгоритма для решения рассматриваемой задачи, с учетом ее размерности.

Также при решении задач необходимо умение выполнять следующие операции для каждой из комбинаторных конфигураций: по имеющейся конфигурации получать следующую за ней в лексикографическом порядке; определять номер данной конфигурации в лексикографической нумерации всех конфигураций; и, наоборот, по порядковому номеру выписывать соответствующую ему конфигурацию.

Литература:

1. Бродский Я. С. Статистика. Вероятность. Комбинаторика / Я. С. Бродский. – М.: ООО «Издательство Оникс»: ООО «Издательство «Мир и Образование», 2008. – 544 с.: ил. – (Школьный курс математики).
2. Семакин И. Г. Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования / И. Г. Семакин, А. П. Шестаков. – М.: Издательский центр «Академия», 2013. – 304 с.

РАЗРАБОТКА СЕРВОТЕСТРА

Мацкевич А.С.,

*обучающийся 3 курса Оршанского колледжа ВГУ имени П.М. Машерова,
г. Орша, Республика Беларусь*

Научный руководитель – Романцов Д.Ю., магистр техн. наук, преподаватель

Введение. Сервотестер – это устройство для тестирования и настройки сервоприводов, которые являются ключевыми компонентами в системах управления механическими устройствами. Сервопривод или сервомашинка – это электродвигатель с блоком управления, который за счёт обратной связи может точно поддерживать заданное положение вала и необходимый угол поворота [1].