

ОСНОВЫ ИСПОЛЬЗОВАНИЯ РЕКУРСИВНЫХ ФУНКЦИЙ В СТРУКТУРАХ ДАННЫХ.

Абдуллаева Мохигул Рахимжон қизи – ассистент кафедры “Информационные технологии, Ферганский филиал Ташкентского Университета информационных технологий имени Мухаммада Аль-Хорезми.

Обухов Вадим Анатольевич – ассистент кафедры “Информационные технологии, Ферганский филиал Ташкентского Университета информационных технологий имени Мухаммада Аль-Хорезми.

Хакимов Ахрор Абдимахаматович – ассистент кафедры “Информационные технологии, Ферганский филиал Ташкентского Университета информационных технологий имени Мухаммада Аль-Хорезми.

Annotatsiya: Maqolada ma'lumotlar tuzilmasi algoritmlari, algoritmlar va uning hossalari, dastur vositalarda qo'llash bo'yicha ma'lumotlar berilgan. Maqolada funksiya va rekursiona funksiya haqida, uning ishlatilishi va iteratsion usullardan afzalligi keltirib o'tilgan, Shuningdek rekursiyani turlari va qo'llash usullari ko'rsatib o'tilgan.

Kalit so'zlar: Algoritm, xossalari, "qonunlar", o'zgarmas takrorlanish munosabatlari, Kompilyator, ro'yxat elementi, rekursiv funksiya.

Аннотация: В статье представлена информация об алгоритмах структуры данных, алгоритмах и их свойствах, применении в программном обеспечении. В статье описаны функция и рекурсивная функция, ее использование и преимущества перед итерационными методами, а также виды рекурсии и методы ее применения.

Ключевые слова: Алгоритм, свойства, «законы», инварианты, рекуррентные соотношения, Компилятор, элемент списка, рекурсивная функция.

Annotation: The article provides information about data structure algorithms, algorithms and their properties, application in software. The article describes a function and a recursive function, its use and advantages over iterative methods, as well as types of recursion and methods for its application.

Key Words: Algorithm, properties, "laws", invariant recurrence relations, Compiler, list element, recursive function.

Введение

Алгоритмы, которые мы до сих пор рассматривали, можно назвать очевидными, их поведение умозрительно наблюдается на образной модели, задача программиста – увидеть его составные части и выстроить их в нужной последовательности. Составные части таких алгоритмов разнородны, а процесс их выполнения можно отследить «исторически». Однако есть и другие подходы

проектированию, кроме образной очевидности. Скажем о них самыми общими словами [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:

- алгоритмы, последовательно приближающиеся к заданной цели;
- алгоритмы, в которых поиск решения сводится к формулировке одной или нескольких задач меньшей размерности;
- алгоритмы, соблюдающие установленные для них соотношения (свойства, «законы», инварианты);

Рекурсия — определение, описание, изображение какого-либо объекта или процесса внутри самого этого объекта или процесса, то есть ситуация, когда объект является частью самого себя. Термин «рекурсия» используется в различных специальных областях знаний — от [лингвистики](#) до [логики](#), но наиболее широкое применение находит в [математике](#) и [информатике](#) [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21].

В [программировании](#) рекурсия — вызов [функции](#) ([процедуры](#)) из неё же самой, непосредственно (*простая рекурсия*) или через другие функции (*сложная* или *косвенная рекурсия*), например, функция вызывает функцию, а функция — функцию. Количество вложенных вызовов функции или процедуры называется глубиной рекурсии. Рекурсивная программа позволяет описать повторяющееся или даже потенциально бесконечное вычисление, причём без явных повторений частей программы и использования циклов. Структурно рекурсивная функция на верхнем уровне всегда представляет собой команду ветвления (выбор одной из двух или более альтернатив в зависимости от условия (условий), которое в данном случае уместно назвать «условием прекращения рекурсии»), имеющую две или более альтернативные ветви, из которых хотя бы одна является *рекурсивной* и хотя бы одна — *терминальной*. Рекурсивная ветвь выполняется, когда условие прекращения рекурсии ложно, и содержит хотя бы один рекурсивный вызов — прямой или опосредованный вызов функцией самой себя. Терминальная ветвь выполняется, когда условие прекращения рекурсии истинно; она возвращает некоторое значение, не выполняя рекурсивного вызова. Правильно написанная рекурсивная функция должна гарантировать, что через конечное число рекурсивных вызовов будет достигнуто выполнение условия прекращения рекурсии, в результате чего цепочка последовательных рекурсивных вызовов прервётся и выполнится возврат. Помимо функций, выполняющих один рекурсивный вызов в каждой рекурсивной ветви, бывают случаи «параллельной рекурсии», когда на одной рекурсивной ветви делается два или более рекурсивных вызова. Параллельная рекурсия типична при обработке сложных структур данных, таких как деревья. Простейший пример параллельно-рекурсивной функции — вычисление ряда Фибоначчи, где для получения значения n -го члена необходимо вычислить $(n-1)$ -й и $(n-2)$ -й.

Реализация рекурсивных вызовов функций в практически применяемых языках и средах программирования, как правило, опирается на механизм [стека вызовов](#) — адрес возврата и локальные переменные функции записываются в [стек](#), благодаря чему каждый следующий рекурсивный вызов этой функции пользуется своим набором локальных переменных и за счёт этого работает корректно. Обратной стороной этого довольно простого по структуре механизма является то, что на каждый рекурсивный вызов требуется некоторое количество [оперативной памяти](#) компьютера, и при чрезмерно большой глубине рекурсии может наступить [переполнение стека](#) вызовов.

Вопрос о желательности использования рекурсивных функций в программировании неоднозначен: с одной стороны, рекурсивная форма может быть структурно проще и нагляднее, в особенности, когда сам реализуемый алгоритм по сути рекурсивен. Кроме того, в некоторых декларативных или чисто функциональных языках (таких как [Пролог](#) или [Haskell](#)) просто нет синтаксических средств для организации циклов, и рекурсия в них — единственный доступный механизм организации повторяющихся вычислений. С другой стороны, обычно рекомендуется избегать рекурсивных программ, которые приводят (или в некоторых условиях могут приводить) к слишком большой глубине рекурсии. Так, широко распространённый в учебной литературе пример рекурсивного вычисления факториала является, скорее, примером того, как *не надо* применять рекурсию, так как приводит к достаточно большой глубине рекурсии и имеет очевидную реализацию в виде обычного циклического алгоритма.[22, 23, 24, 25, 26, 27, 28, 29, 30]

Соответственно, меняется и технология их разработки: основной задачей является – направить алгоритм «в нужное русло», заставить двигаться его в нужном направлении, соблюсти «правила игры», правильно свести задачу к аналогичной. Очевидность и наблюдаемость здесь уступают место логической непротиворечивости и доказательности соблюдения программой заданных свойств[31, 32, 33, 34].

Примеры рекурсии можно обнаружить математике. Например, **рекуррентные соотношения** определяют (вычисляют) некоторый элемент последовательности через несколько предыдущих.

В программировании таких примеров еще больше:

- определение любого конкретного оператора (условный, цикл, блок) в качестве составных частей включает произвольный оператор;
 - рекурсивная структура данных - элемент структуры данных содержит один или несколько указателей на аналогичную структуру данных.
- Например, односвязный список можно определить как элемент списка,

содержащий указатель NULL или указатель на аналогичный список [35, 36, 37];

- рекурсивная функция - тело функции содержит прямой или косвенный (через другую функцию) собственный вызов.

Рекурсивная функция может быть бесконечной, как цикл. Чтобы избежать бесконечного выполнения функции, есть два принципа, которых нужно придерживаться при программировании рекурсивных алгоритмов:

- **Основной критерий** - должно быть, по крайней мере, одно условие, при выполнении которого функция прекращала бы рекурсивный вызов.
- **Прогрессивный подход** - рекурсивные вызовы должны развиваться таким образом, чтобы при каждом вызове приближаться к выполнению базового условия.

Во многих языках программирования рекурсия реализуется с помощью стеков. Всякий раз, когда функция вызывает другую функцию (*или саму себя*), вызывающая функция передает управление вызываемой структуре. Этот процесс передачи также может включать в себя передачу данных от вызывающей функции к вызываемой. Из этого следует, что вызывающая функция должна приостановить свое выполнение, и возобновить его позже, когда управление будет возвращено ей. Вызывающая функция должна запускаться именно в той точке исполнения, где остановилась. Для продолжения работы ей нужны те же значения параметров.

Рекурсию порой сложно понять, особенно новичкам в программировании. Если говорить просто, то рекурсия – это функция, которая сама вызывает себя. Но давайте попробую объяснить на примере. Представьте, что вы пытаетесь открыть дверь в спальню, а она закрыта. Ваш трехлетний сынок появляется из-за угла и говорит, что единственный ключ спрятан в коробке. Вы опаздываете на работу и Вам действительно нужно попасть в комнату и взять вашу рубашку. Вы открываете коробку только чтобы найти... еще больше коробок. Коробки внутри коробок и вы не знаете, в какой из них Ваш ключ. Вам срочно нужна рубашка, так что вам надо придумать хороший алгоритм и найти ключ.

Вычислительная сложность определяется как количество дополнительного объема памяти, необходимого для выполнения модуля. В случае итераций компилятору практически не требуется дополнительной памяти. Компилятор продолжает обновлять значения переменных, используемые в итерациях. Но в случае рекурсивного алгоритма система должна сохранять запись активации каждый раз, когда выполняется рекурсивный вызов. Считается, что вычислительная сложность рекурсивной функции может быть выше, чем у функции с итерацией.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. А. Хакимов. МЕТОДИКА ОЦЕНКИ ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ ERP СИСТЕМ АВТОМАТИЗАЦИИ НА ПРЕДПРИЯТИИ// TATU FF Respublika ilmiy-texnika anjumani -2022 //с- 525-529
2. А. Хакимов SANOAT KORXONALARINING MA'LUMOTLAR BAZALARINI QAYTA ISHLASH TEXNOLOGIK JARAYONLARINI AVTOMATLASHTIRISH// TDTU Respublika miqiyosidagi ilmiy-texnika anjumani// 2021 С-128-129 "
3. Обухов В.А., Горовик А.А., Исследование архитектур и принципов работы современных процессоров / Республиканская научно-техническая конференция по теме «Современные проблемы и решения информационно-коммуникационных технологий и телекоммуникаций». 16-17 апреля 2021 г., ТУИТ ФФ. г. Фергана – с. 217-219.
4. Халилов Д.А., Кушматов О.Э., Обухов В.А., 5 параметров линейки процессоров INTEL: серии, поколения, номера и версии в названии / Республиканская научно-практическая конференция по теме: "Проблемы применения современных информационных, коммуникационных технологий и IT-образования". 24-25 ноября 2021 г., ТУИТ СФ. г. Самарканд – с. 101-105.
5. Обухов В.А. ТУИТ ФФ имени Мухаммада Аль-Хорезми. Диссертационная выпускная работа на тему: "Исследование современных архитектур компьютерных процессоров и разработка компьютерной программы моделирующей работу вычислительных и управляющих узлов процессора". 2022 г.
6. Мохигул А., Мохинур А. ПОНЯТИЕ BIG DATA И ЕГО ОСНОВНЫЕ ХАРАКТЕРИСТИКИ //INTERNATIONAL CONFERENCES ON LEARNING AND TEACHING. – 2022. – Т. 1. – №. 1.
7. Шипулин Ю. Г., Абдуллаев Т. М. Состояние и развитие интеллектуальных оптоэлектронных преобразователей перемещений на основе волоконных и полых световодов //Universum: технические науки. – 2020. – №. 5-1 (74). – С. 5-9.
8. Shipulin Y. et al. Intelligent microprocessor system for control and control of microclimate parameters in vegetable storages using temperature calibrators //Technical science and innovation. – 2021. – Т. 2021. – №. 4. – С. 144-152.
9. Шипулин, Ю. Г., Рустамов, Э., Абдуллаев, Т. М., & Мейлиев, С. Н. (2019). ИНТЕЛЛЕКТУАЛЬНЫЙ ОПТОЭЛЕКТРОННЫЙ ДАТЧИК ТЕМПЕРАТУРЫ С ВОЛОКОННО-ОПТИЧЕСКИМИ ЭЛЕМЕНТАМИ. In

- Проблемы получения, обработки и передачи измерительной информации (pp. 248-253).
10. Shipulin Y. et al. APPLICATION OF METHODS OF INTERMITTENT VENTILATION OF INDUSTRIAL PREMISES USING A DIGITAL DATA TRANSMISSION SYSTEM // Chemical Technology, Control and Management. – 2021. – Т. 2021. – №. 4. – С. 12-18.
 11. Siddikov I. K., Porubay O. V. Neuro-fuzzy system for regulating the processes of power flows in electric power facilities // AIP Conference Proceedings. – AIP Publishing LLC, 2022. – Т. 2432. – №. 1. – С. 020010.
 12. Siddikov I., Porubay O. Neural network model of decision making in electric power facilities under conditions of uncertainty // E3S Web of Conferences. – EDP Sciences, 2021. – Т. 304.
 13. Сиддиков И. Х., Порубай О. В. ПРИНЯТИЕ РЕШЕНИЙ В УСЛОВИЯХ ОПРЕДЕЛЕННОСТИ И РИСКА НА ОСНОВЕ СТРОГИХ МЕТОДОВ // СОВРЕМЕННЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ ФУНДАМЕНТАЛЬНЫХ И ПРИКЛАДНЫХ НАУК. – 2021. – С. 208-214.
 14. Порубай О. В., Амиров А. Р. ПРОБЛЕМЫ ПРИНЯТИЯ РЕШЕНИЙ В УСЛОВИЯХ ОПРЕДЕЛЕННОСТИ И РИСКА НА ОСНОВЕ СТРОГИХ МЕТОДОВ // Universum: технические науки. – 2021. – №. 6-1. – С. 32-33.
 15. Khonturaev, Sardorbek, and Shohida Eshmatova. "Saving environment using Internet of Things: challenges and the possibilities." Современные образовательные технологии в мировом учебно-воспитательном пространстве 8 (2016): 152-157.
 16. А. Хакимов МЕТОДИКА ОЦЕНКИ ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ ЕРСИСТЕМ АВТОМАТИЗАЦИИ НА ПРЕДПРИЯТИИ // TATU FF Respublika ilmiy-texnika anjumani -2022 // с- 525-529
 17. А. Hakimov SANOAT KORXONALARINING MA'LUMOTLAR BAZALARINI QAYTA ISHLASH TEXNOLOGIK JARAYONLARINI AVTOMATLASHTIRISH // TDTU Respublika miqiyosidagi ilmiy-texnika anjumani // 2021 С-128-129 "
 18. Xamidov E. X. MODELS OF OBJECT DETECTION SYSTEM IN VIDEO STREAMS ON A MOBILE DEVICE // Eurasian Journal of Mathematical Theory and Computer Sciences. – 2022. – Т. 2. – №. 3. – С. 21-26.
 19. Khoitkulov, A. A., & Pulatov, G. G. (2022). DEVELOPMENT OF ORGANIZATIONAL AND ECONOMIC MECHANISMS TO INCREASE THE CAPACITY OF TEXTILE ENTERPRISES. *Gospodarka i Innowacje.*, 23, 142-145.

20. Khamidovich X. E., Murodovich X. J. Parallel Programming in Java for Mobile App Development // International Journal of Innovative Analyses and Emerging Technology. – 2022. – Т. 2. – №. 3. – С. 69-74.
21. Khamidovich X. E., Murodovichelnur X. J. Computer-Vision Based Method for Human Action Recognition // International Journal of Innovative Analyses and Emerging Technology. – 2022. – Т. 2. – №. 3. – С. 44-47.
22. Ходжиматов Ж. М. Параллельное программирование в Java // Молодой ученый. – 2021. – №. 22. – С. 30-34.
23. Расулов А. М., Ходжиматов Ж. М. ОБУЧЕНИЕ ПАРАЛЛЕЛЬНОМУ ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ JAVA. – 2021.
24. Khoitkulov A. A. Improving Organizational And Economic Mechanisms To Increase The Power Of Textile Enterprises.
25. M. Sobirov Ta'limda jarayonida LMS tizimlar taxlili // Analytical Journal of Education and Development -2022 // с- 118-122
26. M. Sobirov Advantages of using LMS as a System for Monitoring, Evaluating and Monitoring Learning Outcomes // International Journal of Development and Public Policy // 2022 С-123-128
27. Xamidov Elnur Khamidovich, Xodjimatrov Jahongir Murodovich, 2022/4/2, International Journal of Innovative Analyses and Emerging Technology, 69-74
28. Xamidov Elnur Khamidovich, Xodjimatrov Jahongir Murodovichelnur, 2022/4/1, International Journal of Innovative Analyses and Emerging Technology, 44-47
29. EX Xamidov, 2022/3/24, Eurasian Journal of Mathematical Theory and Computer Sciences, 21-26
30. Эльнур Хамидович Хамидов, 2020, Молодой ученый, 37, 8-11
31. O.I.Ergashev & B.A.Mirzakarimov. Portfolio tizimining tadqiqoti // Central Eurasian Studies Society INTERNATIONAL SCIENTIFIC ONLINE CONFERENCE ON INNOVATION IN THE MODERN EDUCATION SYSTEM collections of scientific works Washington, USA - 2021. Part 13 – №. 3. – С. 399-401.
https://scholar.google.com/citations?view_op=view_citation&hl=ru&user=N8DjLFUAAAAAJ&citation_for_view=N8DjLFUAAAAAJ:2osOgNQ5qMEC
32. O.I.Ergashev & H.Zaynidinov & I.E.Shokirov. Kundalik hayotda sun'iy intellektning eng yaxshi 4 ta misoli // Farg'ona politexnika nstitutida "O'zbekistonda yer yesurklarini boshqarish va ulardan foydalanish tamoyillari: muammo va yechimlar" mavzusida o'tkaziladigan Respublika onlayn ilmiy-amaliy konferensiya 2022, II-tom. – №. 6. – С. 194-199.
https://scholar.google.com/citations?view_op=view_citation&hl=ru&user=N8DjLFUAAAAAJ&citation_for_view=N8DjLFUAAAAAJ:qjMakFHDy7sC

33. O.I.Ergashev & B.A.Mirzakarimov & I.E.Shokirov. Ta'lim muassasalarida avtomatlashtirilgan tizimlarni asosiy tashkil etuvchilari // Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti Farg'ona filiali, "Axborot-kommunikatsiya texnologiyalari va telekommunikatsiyalarning zamonaviy muammolari va yechimlari" Respublika ilmiy-texnik anjumanining ma'ruzalar to'plami. 2019, 30-31 may, III qism – №. 5. – С. 501 – 505
https://scholar.google.com/citations?view_op=view_citation&hl=ru&user=N8DjLFUAAAAAJ&citation_for_view=N8DjLFUAAAAAJ:9yKSN-GCB0IC
34. O.I.Ergashev & H.Zaynidinov & I.E.Shokirov. O'zbekiston Respublikasi o'rta ta'lim o'qituvchilarini portfolio tizimini tadqiqoti va ularni ma'lumotini avtomatlashtirilgan monitoring qilish dasturiy ta'minotini yaratish // POLISH SCIENCE JOURNAL – 2021 may, ISSUE 5(38) Part 2 – №. 3. – С. 117 - 119
https://scholar.google.com/citations?view_op=view_citation&hl=ru&user=N8DjLFUAAAAAJ&citation_for_view=N8DjLFUAAAAAJ:d1gkVwhDpl0C
35. O.I.Ergashev & H.Zaynidinov & I.E.Shokirov. Sun'iy intellekt rivojlanishidagi asosiy to'siqlar // Farg'ona politexnika institutida "O'zbekistonda yer resurslarini boshqarish va ulardan foydalanish tamoyillari: muammo va yechimlar" mavzusida o'tkaziladigan Respublika onlayn ilmiy-amaliy konferensiya - 2022, 23-24 сентябрь, II-том – №. 4. – С. 244 – 247
36. M.Sobirov //Monitoring tizimini avtomatlashtirish jarayoni//Zamonaviy dunyoda ijtimoiy fanlar: nazariy va amaliy izlanishlar//c-2022-115-117
37. M.Sobirov//Issiqlik jarayonlarida energiya tizimini matematik modelning vazifalari//Zamonaviy dunyoda ijtimoiy fanlar: nazariy va amaliy izlanishlar//c-2022-118-122