

Шпаргалка по паттернам алгоритмов

Прочитай задачу и найди её слова во второй колонке. Эта строка и есть твой паттерн. В последней колонке паттерн, с которым его путают, так что проверь его до того, как писать код.

По строке на паттерн, сгруппировано по темам, всего: 52. Каждое название открывает полную страницу с шаблоном на шести языках.

Паттерн	Слова в условии	Время	Не путать с
МАССИВЫ			
Преобразование массива на месте In-place array transform	изменить массив на месте · $O(1)$ дополнительной памяти · не выделять ещё один массив/строку · развернуть/ повернуть/разбить/переместить элементы в одном буфере · вернуть новую длину после удаления дубликатов	$O(n)$	Два указателя (в одну сторону)
ХЕШИРОВАНИЕ			
Хеш-множество / словарь Hash set / map	встречалось ли это раньше · посчитать сколько каждого · найти дубликаты · есть ли подходящее/дополняющее значение · данные в любом порядке / не отсортированы	$O(n)$	Два указателя (с концов)
ДВА УКАЗАТЕЛЯ			
Два указателя (с концов) Two pointers (opposite ends)	отсортированный массив (или можно отсортировать) · найти пару/тройку с заданной суммой · сжать с обоих концов (контейнер, площадь) · проверка палиндрома	$O(n)$	Скользящее окно (переменное)
Два указателя (в одну сторону) Two pointers (same direction)	удалить дубликаты на месте, сохранив порядок · переместить/разбить элементы с сохранением порядка · сжать массив, используя $O(1)$ дополнительной памяти · сдвинуть нули или заданное значение в конец	$O(n)$	Преобразование массива на месте
СКОЛЬЗЯЩЕЕ ОКНО			
Скользящее окно (фиксированный размер) Sliding window (fixed size)	непрерывный подмассив/подстрока заданной длины k · каждое окно размера k · скользящее среднее или скользящая сумма · мин/макс/количество/сумма по каждому фиксированному отрезку	$O(n)$	Скользящее окно (переменное)
Скользящее окно (переменное) Sliding window (variable)	самый длинный/короткий непрерывный подмассив или подстрока · удовлетворяющий условию (сумма $\leq K$, не более, без нарушения лимита) · не более K различных, без повторов, содержит все из X · непрерывный отрезок, не подпоследовательность	$O(n)$	Скользящее окно (фиксированный размер)
ПРЕФИКСНЫЕ СУММЫ			
Префиксные суммы Prefix sums	много запросов на сумму по диапазону · ответить на несколько вопросов о сумме диапазона в одном массиве · сумма от индекса l до r · накопленная или кумулятивная сумма	$O(n)$ build / $O(1)$ query	Скользящее окно (фиксированный размер)
Массив разностей Difference array	сначала много обновлений диапазонов, затем один раз прочитать итоговый массив · прибавить значение к каждому элементу диапазона, повторить много раз · применить к приращений диапазона перед тем как отвечать на запросы · счётчики бронирований/интервалов по массиву	$O(n)$	Префиксные суммы

Паттерн	Слова в условии	Время	Не путать с
ПОИСК			
Линейный поиск Linear search	неотсортированный массив · проверить каждый элемент · нет гарантии порядка · найти первое/любое совпадение · малый n или разовый проход	$O(n)$	Бинарный поиск (массив)
Бинарный поиск (массив) Binary search (array)	отсортированный массив · найти цель / первое или последнее вхождение · n до $1e6+$ и ожидается $O(\log n)$ · найти границу, где условие меняется · поиск во вращённом, но почти отсортированном массиве (с подвохом)	$O(\log n)$	Линейный поиск
Бинарный поиск во вращённом массиве Binary search on a rotated array	отсортированный массив провернут в неизвестной точке · найти цель за $O(\log n)$ · без дубликатов (или обработать их отдельно) · найти минимум / точку вращения · массив растёт, а потом один раз падает	$O(\log n)$	Бинарный поиск (массив)
Бинарный поиск по ответу Binary search on the answer	минимизировать максимум / максимизировать минимум · наименьшее значение, удовлетворяющее условию · найти наименьшую вместимость/скорость/число дней, которое подходит · ответ лежит в числовом диапазоне, а не в позиции массива · проверка выполнимости монотонна (становится true после некоторой точки)	$O(n \log(\text{range}))$	Бинарный поиск (массив)
СОРТИРОВКА			
Элементарные сортировки (выбором, пузырьком, вставками) Elementary sorts (selection, bubble, insertion)	n маленькое (десятки или пара сотен) · массив уже почти отсортирован · сортировка на месте с $O(1)$ дополнительной памяти · учебный/собеседовательный вопрос о механике сортировки · важна стабильность, а простота приемлема	$O(n^2)$	Быстрая сортировка (merge / quick)
Быстрая сортировка (merge / quick) Fast sort (merge / quick)	сортировка массива реального размера (тысячи+) · нужна гарантированная или ожидаемая $O(n \log n)$ · сортировка как подготовительный шаг перед другим алгоритмом (два указателя, жадный, сканирующая прямая) · нужна стабильность (сортировка слиянием) или сортировка на месте (быстрая сортировка) · просто «отсортируй это» без намёка на малое n или особые ключи	$O(n \log n)$	Элементарные сортировки (выбором, пузырьком, вставками)
Сортировка без сравнений (подсчётом / поразрядная) Non-comparison sort (counting / radix)	ключи — целые числа в небольшом известном диапазоне · сортировка возрастов, оценок или ограниченных счётчиков · нужна $O(n)$, а сравнения — узкое место · сортировка по разрядам или ключу фиксированной ширины · n большое, но диапазон значений намного меньше n	$O(n)$	Быстрая сортировка (merge / quick)
Сортировка с пользовательским компаратором Sort with a custom comparator	сортировка по нескольким полям / разрешение ничьей по второму ключу · произвольный порядок (не простое возрастание) · сортировка по убыванию или по вычисляемому ключу · «отсортируй так, чтобы X шло перед Y, если...» · переупорядочивание объектов/записей, а не чисел	$O(n \log n)$	Быстрая сортировка (merge / quick)
ИНТЕРВАЛЫ			
Интервалы: слияние и вставка Intervals: merge & insert	слить перекрывающиеся интервалы · вставить новый интервал в отсортированный список · встречи, брони или диапазоны, которые пересекаются · заданы как пары [начало, конец] · свободное/занятое время между интервалами	$O(n \log n)$	Линия развёртки (подсчёт событий)
Линия развёртки (подсчёт событий) Sweep line (event counting)	максимум одновременно пересекающихся встреч/интервалов · минимум комнат/ресурсов · самый загруженный момент времени · сколько интервалов покрывают заданную точку · события, происходящие в одно и то же время	$O(n \log n)$	Интервалы: слияние и вставка
СВЯЗНЫЕ СПИСКИ			
Разворот связанного списка Linked list reversal	развернуть связный список (весь список или между позициями) · развернуть группами по k · без дополнительного массива или с $O(1)$ доп. памяти · поменять местами пары узлов · односвязный список, доступен только .next	$O(n)$	Слияние и перестройка связанного списка
Быстрый и медленный указатели Fast & slow pointers	обнаружить цикл в связном списке · найти, где начинается цикл · найти средний узел списка · определить, является ли список палиндромом · без доп. памяти / $O(1)$ память, только указатели	$O(n)$	Два указателя (в одну сторону)
Слияние и перестройка связанного списка Linked list merge & reorder	слить два отсортированных связных списка · слить k отсортированных списков · перестроить список (чередовать переднюю и заднюю половины) · переставить узлы без копирования значений в массив · фиктивный головной узел / sentinel	$O(n + m)$	Разворот связанного списка

Паттерн	Слова в условии	Время	Не путать с
СТЕКИ И ОЧЕРЕДИ			
Стек (LIFO) Stack (LIFO)	корректные скобки / сбалансированные скобки · сопоставить самую недавнюю открывающую с ближайшей закрывающей · отмена, откат к последнему состоянию, вложенная структура · вычислить выражение с вложенными операторами · обработать в порядке, обратном порядку поступления	$O(n)$	Монотонный стек
Монотонный стек Monotonic stack	ближайший больший / ближайший меньший элемент · сколько дней ждать потепления · наибольший прямоугольник в гистограмме · сколько воды соберётся между столбиками · отрезок/серия, заканчивающаяся первым большим или меньшим значением	$O(n)$	Стек
Монотонная дек-очередь (максимум/минимум окна) Monotonic deque (sliding window max/min)	максимум или минимум каждого окна размера K · максимум/минимум скользящего окна · кратчайший подмассив с суммой не менее K · нужно снимать/класть с обоих концов очереди · дек, двусторонняя очередь	$O(n)$	Монотонный стек
РЕКУРСИЯ			
Рекурсия Recursion	задача определена через меньшую версию самой себя · вложенная/древовидная структура для обхода (папки, выражение, связный список) · нужно посчитать сумму/количество/глубину по всем подэлементам · естественное деление на базовый случай и рекурсивный (факториал, Фибоначчи, глубина дерева)	$O(n)$	Рекурсия с мемоизацией
Рекурсия с мемоизацией Recursion with memoization	рекурсивные вызовы повторяют одни и те же аргументы (пересекающиеся подзадачи) · обычная рекурсия работает верно, но слишком медленно / экспоненциальный взрыв · мало различных состояний, хотя дерево рекурсии огромное · закешировать результат ($i, \dots$) перед возвратом · формулировка вроде Фибоначчи / подъём по лестнице / путь по сетке с намёком на кеш	$O(\text{states})$	Рекурсия
ПЕРЕБОР С ВОЗВРАТОМ			
Бэктрекинг Backtracking	сгенерировать все подмножества/перестановки/комбинации · n маленькое (примерно $n \leq 12-20$), экспоненциальный перебор допустим · сделать выбор, уйти в рекурсию, затем отменить выбор перед следующим · расставить элементы с ограничениями и откатиться при конфликте (N ферзей, sudoku) · вернуть все допустимые варианты, а не один	$O(2^n)/O(n!)$	Рекурсия
ДЕРЕВЬЯ			
Обход дерева Tree traversal	бинарное дерево задано как корневой узел с указателями left/right · посетить каждый узел (прямой / симметричный / обратный / по уровням) · посчитать глубину, высоту или сумму по всему дереву · обработать детей до или после родителя · вывод по уровням (BFS с очередью)	$O(n)$	Обход графа BFS / DFS
Двоичное дерево поиска Binary search tree	двоичное дерево поиска: $\text{left} < \text{node} < \text{right}$ в каждом узле · вставка/удаление/поиск с сохранением порядка данных · проверить BST / k -й наименьший / симметричный обход даёт отсортированную последовательность · ближайшее значение или запрос по диапазону на множестве, которое меняется со временем · массив заранее не дан, значения приходят по одному	$O(\log n)$ avg	Бинарный поиск (массив)
Наименьший общий предок Lowest common ancestor	наименьший общий предок двух узлов · самый глубокий узел, который является предком и p , и q · каждый рекурсивный вызов возвращает, какой из целевых узлов найден под ним · путь от корня до каждого узла и их сравнение (альтернативный подход) · дано бинарное дерево или BST, найти точку, где расходятся два пути	$O(n)$	Двоичное дерево поиска
Сериализация / десериализация дерева Tree serialize / deserialize	сериализовать дерево в строку и десериализовать обратно · восстановить точно ту же форму, а не только те же значения · формулировка encode/decode, разработать кодек для дерева · маркеры null/None для отсутствующих детей · сохранить или передать дерево (по сети, в файл, в кеш)	$O(n)$	Обход дерева
КУЧИ И ТОП-К			
Бинарная куча / очередь с приоритетом Binary heap / priority queue	всегда нужен текущий минимум/максимум · k наименьших/наибольших элементов · слияние k отсортированных списков · текущая медиана, пока элементы прибывают · очередь с приоритетом / самое срочное следующим	$O(\log n)$ per op	Быстрая сортировка (merge / quick)

Паттерн	Слова в условии	Время	Не путать с
ГРАФЫ			
Обход графа BFS / DFS Graph BFS / DFS	кратчайший путь в невзвешенном графе · минимальное число шагов/рёбер между узлами · достижим ли узел Y из узла X · исследовать кольцо за кольцом / слой за слоем · граф задан списком смежности, возможны циклы	$O(V+E)$	Обход дерева
Компоненты связности Connected components	посчитать число отдельных групп/островов/кластеров · граф может быть несвязным · сколько связанных частей · круги друзей · разметить связанные области	$O(V+E)$	Обход графа BFS / DFS
Топологическая сортировка (алгоритм Кана) Topological sort (Kahn's algorithm)	задача должна идти перед другой · порядок, соблюдающий зависимости · обнаружить цикл в предпосылках · порядок сборки / расписание курсов / порядок установки · ориентированный граф, X должен произойти перед Y	$O(V+E)$	Обход графа BFS / DFS
Система непересекающихся множеств Union-find (disjoint set)	лежат ли эти двое в одной группе · факты о слиянии приходят по одному · связность меняется по мере добавления рёбер · склейка аккаунтов или друзей во времени · это ребро замкнёт цикл	near $O(1)$ per op	Компоненты связности
Алгоритм Дейкстры Dijkstra's algorithm	самый дешёвый/короткий путь со взвешенными рёбрами · неотрицательные веса рёбер · минимальная суммарная стоимость до каждого узла · задержка в сети / стоимость маршрута, а не число дорог · взвешенный граф, кратчайшие пути из одного источника	$O(E \log V)$	Обход графа BFS / DFS
Эйлеров путь (алгоритм Хирхольцера) Eulerian path (Hierholzer's algorithm)	использовать каждое ребро ровно один раз · восстановить маршрут по билетам · рисунок одним росчерком / не отрывая ручку · пройти каждую дорогу/домино ровно один раз · рейсы или домино должны соединиться в одну поездку	$O(E)$	Обход графа BFS / DFS
Проверка двудольности (раскраска в два цвета) Bipartite check (two-coloring)	разбить на две группы без конфликта внутри группы · раскрасить граф в два цвета · обнаружить цикл нечётной длины · мы против них / расписание на две смены / распределение по двум комнатам · возможно ли двудольное разбиение конфликтующих пар	$O(V+E)$	Обход графа BFS / DFS
ЖАДНЫЕ АЛГОРИТМЫ			
Жадный алгоритм (обменный аргумент) Greedy (exchange argument)	наименьшее число монет или частей на сумму · берём самую крупную, что ещё помещается · выдать сдачу номиналами · доказать оптимальность обменным аргументом	$O(n \log n)$	Динамическое программирование (1-D)
ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ			
Динамическое программирование (1-D) Dynamic programming (1-D)	лучшее/максимум/минимум, заканчивающееся в позиции i · число способов дойти, подняться или замостить · нельзя брать два соседних элемента · минимальная стоимость пути по сетке · повторяющиеся подзадачи, одно и то же состояние считается заново	$O(\text{states})$	Рекурсия с мемоизацией
DP над подпоследовательностями (LIS / LCS / расстояние редактирования) DP on subsequences (LIS / LCS / edit distance)	самая длинная возрастающая подпоследовательность · самая длинная общая подпоследовательность двух строк · минимум правок, чтобы превратить одну строку в другую · сохранить относительный порядок, можно пропускать элементы · diff или выравнивание двух последовательностей	$O(n^2)/O(nm)$	Динамическое программирование (1-D)
DP на интервалах DP on intervals	объединить соседние части с некоторой стоимостью · лучший способ разбить отрезок на две части · порядок перемножения цепочки матриц · стоимость слияния зависит от всего отрезка · оптимальная расстановка скобок или порядок построения	$O(n^3)$	Динамическое программирование (1-D)
DP на деревьях DP on trees	лучший выбор по иерархии или оргструктуре · нельзя выбрать узел вместе с его прямым родителем · постпорядок: дети решаются раньше родителя · каждый узел объединяет состояния своих детей	$O(n)$	Обход дерева
СТРОКИ И ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ			
DP на строках (word break) DP on strings (word break)	можно ли разбить строку на слова из словаря · сегментировать строку по списку слов · булево значение достижимости в каждой точке разреза · слитный текст нужно разделить на границы слов	$O(n^2)$	DP над подпоследовательностями (LIS / LCS / расстояние редактирования)
СТРОКИ			
Trie (префиксное дерево) Trie (prefix tree)	автодополнение или typeahead по множеству строк · самый длинный общий префикс среди слов · вставка, поиск и проверка префикса в наборе слов · сопоставление с большим фиксированным словарём по префиксу	$O(\text{len})$ per op	Хеш-множество / словарь
БИТОВЫЕ ОПЕРАЦИИ			
Битовые манипуляции Bit manipulation	найти единственное или уникальное число через XOR · установить, снять, переключить или проверить бит · упаковать много флагов да/нет в одно целое число · подсчёт установленных битов или проверка степени двойки	$O(1)/O(n)$	Хеш-множество / словарь

Паттерн	Слова в условии	Время	Не путать с
МАТЕМАТИКА И ТЕОРИЯ ЧИСЕЛ			
Математика и теория чисел (НОД, решето, модульная арифметика) Math and number theory (GCD, sieve, modular)	сократить дробь до несократимого вида · перечислить все простые числа до n · ответ по модулю $1e9+7$ · когда снова совпадут два повторяющихся интервала (НОК)	varies	Битовые манипуляции
МАТРИЦЫ			
Обход матрицы (спираль / диагонали) Matrix traversal (spiral / diagonal)	прочитать или заполнить сетку по спирали · обойти матрицу по диагоналям · повернуть матрицу на 90 градусов · посетить каждую ячейку в фиксированном геометрическом порядке	$O(mn)$	Поиск в матрице (отсортированная сетка)
Поиск в матрице (отсортированная сетка) Matrix search (sorted grid)	матрица отсортирована и по строкам, и по столбцам · поиск значения в 2D отсортированной сетке · начать с угла и отбрасывать строку или столбец · и строки, и столбцы монотонны	$O(m+n)$	Бинарный поиск (массив)
Поиск слова в матрице (DFS/backtracking по сетке) Matrix word search (grid DFS/backtracking)	существует ли слово как путь по доске · проследить буквы через соседние клетки без повторов · DFS с возвратом (backtracking) по сетке · пометить клетку посещённой, затем отменить пометку	$O(mn*4^L)$	Бэктрекинг
ПРОДВИНУТЫЕ ТЕМЫ			
Дерево Фенвика / дерево отрезков Fenwick tree / segment tree	сумма или запрос на отрезке с точечными обновлениями вперемешку · живой лидерборд или ранг, пока баллы меняются · подсчёт инверсий или сколько более ранних значений больше · запросы на отрезке, пока данные продолжают меняться	$O(\log n)$ per op	Префиксные суммы