

Оглавление

Предисловие	14
О чем эти книги.....	14
Навыки, которые вы приобретете.....	16
В чем особенность этой книги	17
Для кого эта книга?.....	18
Дополнительные ресурсы	19
Благодарности.....	20
Глава 1. Введение	21
1.1. Зачем изучать алгоритмы?.....	22
1.2. Целочисленное умножение	24
1.2.1. Задачи и решения.....	24
1.2.2. Задача целочисленного умножения	25
1.2.3. Алгоритм начальной школы	25
1.2.4. Анализ числа операций.....	27
1.2.5. Можно ли добиться лучшего?	27
1.3. Умножение Карацубы	28
1.3.1. Конкретный пример	28
1.3.2. Рекурсивный алгоритм.....	30
1.3.3. Умножение Карацубы	32
1.4. Алгоритм MergeSort	36
1.4.1. Актуальность	36
1.4.2. Сортировка	37

1.4.3. Пример	39
1.4.4. Псевдокод.....	40
1.4.5. Подпрограмма Merge	41
1.5. Анализ алгоритма MergeSort	42
1.5.1. Время исполнения подпрограммы Merge.....	43
1.5.2. Время исполнения алгоритма MergeSort.....	44
1.5.3. Доказательство теоремы 1.2.....	46
1.5.4. Ответы на тестовые задания 1.1–1.2.....	50
Ответ на тестовое задание 1.1	50
Ответ на тестовое задание 1.2	51
1.6. Основные принципы анализа алгоритмов	51
1.6.1. Принцип № 1: анализ наихудшего случая.....	52
1.6.2. Принцип № 2: анализ значимых деталей.....	53
1.6.3. Принцип № 3: асимптотический анализ.....	55
1.6.4. Что такое «быстрый» алгоритм?	58
Задачи на закрепление материала	60
Задача повышенной сложности	62
Задача по программированию	62

Глава 2. Асимптотические обозначения 63

2.1. Отправная точка.....	64
2.1.1. Актуальность	64
2.1.2. Высокоуровневая идея.....	65
2.1.3. Четыре примера	67
2.1.4. Решения тестовых заданий 2.1–2.4	72
2.2. Обозначение O -большое.....	74
2.2.1. Определение на русском языке.....	74
2.2.2. Иллюстрированное определение	74
2.2.3. Математическое определение.....	75
2.3. Два простых примера	77
2.3.1. Для многочленов степени k O -большое является $O(n^k)$	77
2.3.2. Для многочленов степени k O -большим не является $O(n^{k-1})$	79

2.4.	Обозначение Омега-большое и Тета-большое.....	80
2.4.1.	Обозначение Омега-большое	80
2.4.2.	Обозначение Тета-большое	81
2.4.3.	Обозначение о-малое	83
2.4.4.	Откуда взялось обозначение?	84
2.4.5.	Решение тестового задания 2.5.....	84
2.5.	Дополнительные примеры	85
2.5.1.	Добавление константы к экспоненте.....	85
2.5.2.	Умножение экспоненты на константу	86
2.5.3.	Максимум против суммы	87
Задачи	на закрепление материала	89

Глава 3. Алгоритмы «разделяй и властвуй» 91

3.1.	Парадигма «разделяй и властвуй».....	92
3.2.	Подсчет инверсий за время $O(n \log n)$	93
3.2.1.	Задача	93
3.2.2.	Пример	93
3.2.3.	Совместная фильтрация	94
3.2.4.	Поиск полным перебором	95
3.2.5.	Подход «разделяй и властвуй»	96
3.2.6.	Высокоуровневый алгоритм	96
3.2.7.	Ключевая идея: задействовать алгоритм MergeSort	97
3.2.8.	К вопросу о подпрограмме Merge.....	99
3.2.9.	Подпрограмма Merge и разделенные инверсии	100
3.2.10.	Подпрограмма Merge-CountSplitInv	102
3.2.11.	Корректность	103
3.2.12.	Время исполнения.....	103
3.2.13.	Решения тестовых заданий 3.1 – 3.2	104
3.3.	Умножения матриц по алгоритму Штрассена	104
3.3.1.	Умножение матриц	105
3.3.2.	Пример ($n = 2$)	105
3.3.3.	Простой алгоритм	106

3.3.4. Подход «разделяй и властвуй»	107
3.3.5. Экономия времени на рекурсивном вызове	109
3.3.6. Детали	111
3.3.7. Решение тестового задания 3.3	112
*3.4. Алгоритм со временем $O(n \log n)$ для ближайшей пары	112
3.4.1. Задача	113
3.4.2. Разминка: одномерный случай	113
3.4.3. Предварительная обработка	114
3.4.4. Подход «разделяй и властвуй»	116
3.4.5. Тонкая настройка	118
3.4.6. Подпрограмма ClosestSplitPair	119
3.4.7. Правильность	121
3.4.8. Доказательство леммы 3.3 (а)	122
3.4.9. Доказательство леммы 3.3 (b)	123
3.4.10. Решение тестового задания 3.4	125
Задача на закрепление материала	126
Задачи повышенной сложности	127
Задачи по программированию	128
Глава 4. Основной метод	129
4.1. К вопросу о целочисленном умножении	130
4.1.1. Алгоритм RecIntMult	130
4.1.2. Алгоритм Karatsuba	131
4.1.3. Сравнение рекуррентных соотношений	132
4.2. Формулировка	133
4.2.1. Стандартные рекуррентные соотношения	133
4.2.2. Формулировка и обсуждение основного метода	135
4.3. Шесть примеров	137
4.3.1. К вопросу об алгоритме MergeSort	137
4.3.2. Двоичный поиск	138
4.3.3. Рекурсивное целочисленное умножение	139
4.3.4. Умножение Карацубы	139

4.3.5. Умножение матриц	140
4.3.6. Фиктивное рекуррентное соотношение	141
4.3.7. Решения тестовых заданий 4.2–4.3	142
*4.4. Доказательство основного метода.....	143
4.4.1. Преамбула	144
4.4.2. К вопросу о деревьях рекурсии.....	145
4.4.3. Работа, выполняемая на одном уровне	146
4.4.4. Суммирование уровней	147
4.4.5. Добро против зла: потребность в трех случаях.....	148
4.4.6. Предсказание границ времени работы.....	149
4.4.7. Заключительные расчеты: случай 1	151
4.4.8. Отклонение: геометрический ряд	151
4.4.9. Заключительные вычисления: случаи 2 и 3.....	152
4.4.10. Решения тестовых заданий 4.4–4.5	153
Задачи на закрепление материала	156
Задача повышенной сложности	157
Глава 5. Алгоритм QuickSort	158
5.1. Обзор	159
5.1.1. Сортировка	159
5.1.2. Разделение вокруг опорного элемента.....	160
5.1.3. Высокоуровневое описание.....	162
5.1.4. Забегая вперед	163
5.2. Разделение массива вокруг опорного элемента	164
5.2.1. Легкий выход из положения	164
5.2.2. Реализация на том же месте: высокоуровневый план	165
5.2.3. Пример	166
5.2.4. Псевдокод Partition.....	169
5.2.5. Псевдокод алгоритма Quicksort.....	171
5.3. Важность хороших опорных элементов	171
5.3.1. Наивная реализация подпрограммы ChoosePivot.....	172

5.3.2. Избыточная реализация ChoosePivot.....	173
5.3.3. Решения тестовых заданий 5.1– 5.2.....	174
5.4. Рандомизированный алгоритм Quicksort.....	176
5.4.1. Рандомизированная реализация подпрограммы ChoosePivot.....	176
5.4.2. Время работы рандомизированного алгоритма Quicksort.....	177
5.4.3. Интуитивное понимание: чем хороши случайные опорные элементы?.....	178
*5.5. Анализ рандомизированного Quicksort.....	180
5.5.1. Предварительные сведения.....	181
5.5.2. Схема декомпозиции.....	183
5.5.3. Применение схемы.....	185
5.5.4. Вычисление вероятностей сравнения.....	187
5.5.5. Заключительные вычисления.....	190
5.5.6. Решение тестового задания 5.3.....	192
*5.6. Сортировка требует $\Omega(n \log n)$ сравнений.....	192
5.6.1. Алгоритмы сортировки на основе сравнения.....	193
5.6.2. Более быстрая сортировка при более строгих допущениях.....	194
5.6.3. Доказательство теоремы 5.5.....	196
Задачи на закрепление материала.....	199
Задача повышенной сложности.....	201
Задачи по программированию.....	201
Глава 6. Линейный выбор.....	203
6.1. Алгоритм RSelect.....	204
6.1.1. Задача выбора.....	204
6.1.2. Сведение к задаче сортировки.....	206
6.1.3. Подход «разделяй и властвуй».....	207
6.1.4. Псевдокод для алгоритма RSelect.....	208
6.1.5. Время работы алгоритма RSelect.....	209
6.1.6. Решение тестовых заданий 6.1–6.2.....	212
Решение тестового задания 6.1.....	212
Решение тестового задания 6.2.....	212

*6.2. Анализ алгоритма RSelect	213
6.2.1. Отслеживание прогресса посредством фаз	213
6.2.2. Сведение к задаче подбрасывания монеты	215
6.2.3. Соединяем все вместе.....	217
*6.3. Алгоритм DSelect.....	218
6.3.1. Гениальная идея: медиана медиан.....	219
6.3.2. Псевдокод для алгоритма DSelect	220
6.3.3. Понимание алгоритма DSelect.....	221
6.3.4. Время работы алгоритма DSelect.....	222
*6.4. Анализ алгоритма DSelect	224
6.4.1. Работа вне рекурсивных вызовов.....	224
6.4.2. Грубое рекуррентное соотношение	225
6.4.3. Лемма 30–70.....	226
6.4.4. Решение рекуррентного соотношения.....	228
6.4.5. Метод догадок и проверок	230
Задачи на закрепление материала	233
Задачи повышенной сложности	234
Задачи по программированию	234

Приложения 235

Приложение А. Краткий обзор доказательств по индукции.....	236
А.1. Шаблон для доказательств по индукции.....	236
А.2. Пример: замкнутая формула.....	237
А.3. Пример: размер полного двоичного дерева	238
Приложение Б. Краткий обзор дискретной вероятности	239
Б.1. Выборочные пространства.....	240
Б.2. События	241
Б.3. Случайные величины.....	243
Б.4. Математическое ожидание	244
Б.5. Линейность математического ожидания	247
Б.6. Пример: распределение нагрузки.....	250