

Содержание

Поиск...

Время в...	30
Наглядн...	33
«О-боль...	35
Типичн...	36
Упражнения	37
Задача ...	38
Шпаргалка	40
Глава 2. Сорти...	41
Как работа...	42
Массивы и ...	44
Связанн...	46
Массивы	47
Термин...	48
Упражнения	49
Вставка ...	50
Удаление	51
Упражнения	52
Сортировк...	54
Пример кода	58
Шпаргалка	59
Глава 3. Рекур...	60
Рекурсия	61
Базовый сл...	64
Стек	66

ЛОГАРИФМЫ

Возможно, вы уже забыли, что такое логарифм, но наверняка помните, что такое возведение в степень. $\log_{10} 100$ по сути означает, сколько раз нужно перемножить 10, чтобы получить 100. Правильный ответ — 2: 10×10 . Итак, $\log_{10} 100 = 2$. Логарифм по смыслу противоположен возведению в степень.

$$10^2 = 100 \leftrightarrow \log_{10} 100 = 2$$

$$10^3 = 1000 \leftrightarrow \log_{10} 1000 = 3$$

$$2^3 = 8 \leftrightarrow \log_2 8 = 3$$

$$2^4 = 16 \leftrightarrow \log_2 16 = 4$$

$$2^5 = 32 \leftrightarrow \log_2 32 = 5$$



Шпаргалка

- ❑ Память компьютера напоминает огромный шкаф с ящиками.
- ❑ Если вам потребуется сохранить набор элементов, воспользуйтесь массивом или списком.
- ❑ В массиве все элементы хранятся в памяти рядом друг с другом.
- ❑ В списке элементы распределяются в произвольных местах памяти, при этом в одном элементе хранится адрес следующего элемента.
- ❑ Массивы обеспечивают быстрое чтение.
- ❑ Списки обеспечивают быструю вставку и выполнение.
- ❑ Все элементы массива должны быть однотипными (только целые числа, только вещественные числа и т. д.).

Содержание

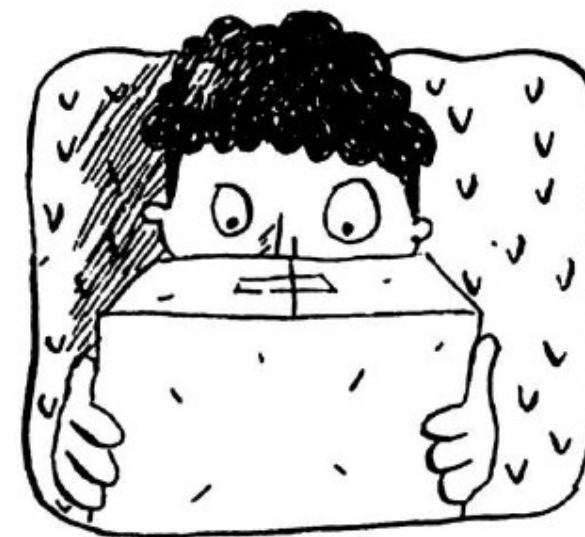
Поиск...

Структура к...	16
Как работа...	17
Для кого пр...	17
Условные о...	18
Об авторе	18
От издател...	18
Глава 1. Знако...	19
Введение	19
Что вы узн...	20
Что вы узн...	20
Бинарный ...	21
Более э...	24
Упражнения	28
Время в...	29
«О-большое»	30
Время в...	30
Наглядн...	33
«О-боль...	35
Типичн...	36
Упражнения	37
Задача ...	38
Шпаргалка	40
Глава 2. Сорти...	41
Как работа...	42
Массивы и ...	44
Связанн...	46



Шпаргалка

- ❑ Когда функция вызывает саму себя, это называется рекурсией.
- ❑ В каждой рекурсивной функции должно быть два случая: базовый и рекурсивный.
- ❑ Стек поддерживает две операции: занесение и извлечение элементов.
- ❑ Все вызовы функций сохраняются в стеке вызовов.
- ❑ Если стек вызовов станет очень большим, он займет слишком много памяти.



Содержание

Поиск...

Глава 1. Знакомство	19
Введение	19
Что вы узнаете	20
Что вы узнаете	20
Бинарный поиск	21
Более эффективно	24
Упражнения	28
Время выполнения	29
«О-большое»	30
Время выполнения	30
Наглядность	33
«О-большое»	35
Типичные ошибки	36
Упражнения	37
Задача	38
Шпаргалка	40
Глава 2. Сортировка	41
Как работает	42
Массивы и указатели	44
Связанные массивы	46
Массивы	47
Термин	48
Упражнения	49
Вставка	50
Удаление	51
Упражнения	52


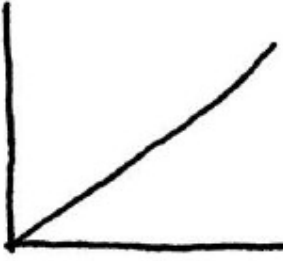





«О-большое».

Содержание

Поиск...

Структура к...	16
Как работа...	17
Для кого пр...	17
Условные о...	18
Об авторе	18
От издател...	18
Глава 1. Знако...	19
Введение	19
Что вы узн...	20
Что вы узн...	20
Бинарный ...	21
Более э...	24
Упражнения	28
Время в...	29
«О-большое»	30
Время в...	30
Наглядн...	33
«О-боль...	35
Типичн...	36
Упражнения	37
Задача ...	38
Шпаргалка	40
Глава 2. Сорти...	41
Как работа...	42
Массивы и ...	44
Связанн...	46

ПРИМЕР АЛГОРИТМА:	БИНАРНЫЙ ПОИСК	ПРОСТОЙ ПОИСК	БЫСТРАЯ СОРТИРОВКА	СОРТИРОВКА ВЫБОРОМ	ЗАДАЧА О КОМ- МИВОЯЖЕРЕ
					
РАЗМЕР МАССИВА	$O(\log n)$	$O(n)$	$O(n \log n)$	$O(n^2)$	$O(n!)$
10	0.3 с	1 с	3.3 с	10 с	4.2 дня
100	0.6 с	10 с	664 с	16.6 мин	2.9×10^{149} ЛЕТ
1000	1 с	100 с	996 с	27.7 час	1.27×10^{2559} ЛЕТ

Оценки для медленного компьютера, выполняющего 10 операций в секунду



Системный мон...



РАСПИСАНИЕ - М...



Бхаргава - Грока...



РАСПИСАНИЕ : ba...

08:53
ЧТ, 24 НОЯ

Содержание

Поиск...

Типичн...	36
Упражнения	37
Задача ...	38
Шпаргалка	40
Глава 2. Сорти...	41
Как работа...	42
Массивы и ...	44
Связанн...	46
Массивы	47
Термин...	48
Упражнения	49
Вставка ...	50
Удаление	51
Упражнения	52
Сортировк...	54
Пример кода	58
Шпаргалка	59
Глава 3. Рекур...	60
Рекурсия	61
Базовый сл...	64
Стек	66
Стек вы...	67
Упражнения	69
Стек вы...	70
Упражнения	74
Шпаргалка	75

Шпаргалка

- ❑ Жадные алгоритмы стремятся к локальной оптимизации в расчете на то, что в итоге будет достигнут глобальный оптимум.
- ❑ У NP-полных задач не существует известных быстрых решений.
- ❑ Если у вас имеется NP-полная задача, лучше всего воспользоваться приближенным алгоритмом.
- ❑ Жадные алгоритмы легко реализуются и быстро выполняются, поэтому из них получаются хорошие приближенные алгоритмы.



Содержание

Поиск...

Предисловие	12
Благодарности	13
О книге	15
Структура к...	16
Как работа...	17
Для кого пр...	17
Условные о...	18
Об авторе	18
От издател...	18
Глава 1. Знако...	19
Введение	19
Что вы узн...	20
Что вы узн...	20
Бинарный ...	21
Более э...	24
Упражнения	28
Время в...	29
«О-большое»	30
Время в...	30
Наглядн...	33
«О-боль...	35
Типичн...	36
Упражнения	37
Задача ...	38
Шпаргалка	40
Глава 2. Сорти...	41

в разные цвета. Убедитесь ли эта задача NP-полной?

Шпаргалка

- ❑ Жадные алгоритмы стремятся к локальной оптимизации в расчете на то, что в итоге будет достигнут глобальный оптимум.
- ❑ У NP-полных задач не существует известных быстрых решений.
- ❑ Если у вас имеется NP-полная задача, лучше всего воспользоваться приближенным алгоритмом.
- ❑ Жадные алгоритмы легко реализуются и быстро выполняются, поэтому из них получаются хорошие приближенные алгоритмы.



- ❑ Динамическое программирование применяется при оптимизации некоторой характеристики.

- ❑ Динамическое программирование работает только в ситуациях, в которых задача может быть разбита на автономные подзадачи.
- ❑ В каждом решении из области динамического программирования строится таблица.
- ❑ Значения ячеек таблицы обычно соответствуют оптимизируемой характеристике.
- ❑ Каждая ячейка представляет подзадачу, поэтому вы должны подумать о том, как разбить задачу на подзадачи.
- ❑ Не существует единой формулы для вычисления решений методом динамического программирования.

Содержание

Поиск...

Предисловие	12
Благодарности	13
О книге	15
Структура к...	16
Как работа...	17
Для кого пр...	17
Условные о...	18
Об авторе	18
От издател...	18
Глава 1. Знако...	19
Введение	19
Что вы узн...	20
Что вы узн...	20
Бинарный ...	21
Более э...	24
Упражнения	28
Время в...	29
«О-большое»	30
Время в...	30
Наглядн...	33
«О-боль...	35
Типичн...	36
Упражнения	37
Задача ...	38
Шпаргалка	40
Глава 2. Сорти...	41

- ❑ Алгоритм k ближайших соседей применяется для классификации и регрессии. В нем используется проверка k ближайших соседей.

Шпаргалка **253**

- ❑ Классификация = распределение по категориям.
- ❑ Регрессия = прогнозирование результата (например, в виде числа).
- ❑ «Извлечением признаков» называется преобразование элемента (например, фрукта или пользователя) в список чисел, которые могут использоваться для сравнения.
- ❑ Качественный выбор признаков — важная часть успешного алгоритма k ближайших соседей.



Содержание

Поиск...

Время в...	30
Наглядн...	33
«О-боль...	35
Типичн...	36
Упражнения	37
Задача ...	38
Шпаргалка	40
Глава 2. Сорти...	41
Как работа...	42
Массивы и ...	44
Связанн...	46
Массивы	47
Термин...	48
Упражнения	49
Вставка ...	50
Удаление	51
Упражнения	52
Сортировк...	54
Пример кода	58
Шпаргалка	59
Глава 3. Рекур...	60
Рекурсия	61
Базовый сл...	64
Стек	66

$$2^5 = 32 \leftrightarrow \log_2 32 = 5$$

Логарифм — операция, обратная возведению в степень

Когда я в этой книге упоминаю «О-большое» (об этом чуть позднее), \log всегда означает \log_2 . Когда вы ищете элемент с применением простого поиска, в худшем случае вам придется проверить каждый элемент. Итак, для списка из 8 чисел понадобится не больше 8 проверок. Для бинарного поиска в худшем случае потребуется не более \log_n проверок. Для списка из 8 элементов $\log_8 == 3$, потому что $2^3 == 8$. Итак, для списка из 8 чисел вам придется проверить не более 3 чисел. Для списка из 1024 элементов $\log 1024 = 10$, потому что $2^{10} == 1024$. Следовательно, для списка из 1024 чисел придется проверить не более 10 чисел.



работу).

Шпаргалка

Хеши хорошо подходят для решения следующих задач:

- ☐ моделирование отношений между объектами;
- ☐ устранение дубликатов;
- ☐ кэширование/запоминание данных вместо выполнения работы на сервере.

Коллизии

Как я уже сказал, в большинстве языков существуют свои хеш-таблицы. Вам не нужно знать, как написать собственную реализацию, поэтому я не буду надолго останавливаться на внутреннем строении хеш-таблиц. Но быстродействие тоже важно! Чтобы понять, быстродействие хеш-



Содержание

Упражнения	28
Время в...	29
«О-большое»	30
Время в...	30
Наглядн...	33
«О-боль...	35
Типичн...	36
Упражнения	37
Задача ...	38
Шпаргалка	40
Глава 2. Сорти...	41
Как работа...	42
Массивы и ...	44
Связанн...	46
Массивы	47
Термин...	48
Упражнения	49
Вставка ...	50
Удаление	51
Упражнения	52
Сортировк...	54
Пример кода	58
Шпаргалка	59
Глава 3. Рекур...	60
Рекурсия	61
Базовый сл...	64

