timyrik20 29 июля 2013 в 00:00

# Знай сложности алгоритмов

Автор оригинала: Eric Rowell

Алгоритмы

Перевод

Эта статья рассказывает о времени выполнения и о расходе памяти большинства алгоритмов
прошлом, когда я готовился к прохождению собеседования я потратил много времени иссле,
информации о лучшем, среднем и худшем случае работы алгоритмов поиска и сортировки, ч
собеседовании не поставил меня в тупик. За последние несколько лет я проходил интервью в
Силиконовой долины, а также в некоторых крупных компаниях таких как Yahoo, eBay, LinkedI
готовился к интервью, я подумал: «Почему никто не создал хорошую шпаргалку по асимптоті
Чтобы сохранить ваше время я создал такую шпаргалку. Наслаждайтесь!

| Хорошо | Приемлемо | Плохо |
|---|---|---|

## Поиск

| Алгоритм | Структура данных | Времен |
|---|---|---|
| | | В среднем |
| Поиск в глубину (DFS) | Граф с \|V\| вершинами и \|E\| ребрами | - |
| Поиск в ширину (BFS) | Граф с \|V\| вершинами и \|E\| ребрами | - |
| Бинарный поиск | Отсортированный массив из n элементов | O(log(n)) |
| Линейный поиск | Массив | O(n) |
| Кратчайшее расстояние по алгоритму Дейкстры используя двоичную кучу как очередь с приоритетом | Граф с \|V\| вершинами и \|E\| ребрами | O((\|V\| + \|E\|) log \|V |
| Кратчайшее расстояние по алгоритму Дейкстры используя массив как очередь с приоритетом | Граф с \|V\| вершинами и \|E\| ребрами | O(\|V\|^2) |
| Кратчайшее расстояние используя алгоритм Беллмана–Форда | Граф с \|V\| вершинами и \|E\| ребрами | O(\|V\|\|E\|) |

## Сортировка

| Алгоритм | Структура данных | Временная сложность | | |
|---|---|---|---|---|
| | | Лучшее | В среднем | В худшем |
| Быстрая сортировка | Массив | O(n log(n)) | O(n log(n)) | O(n^2) |
| Сортировка слиянием | Массив | O(n log(n)) | O(n log(n)) | O(n log(n)) |
| Пирамидальная сортировка | Массив | O(n log(n)) | O(n log(n)) | O(n log(n)) |
| Пузырьковая сортировка | Массив | O(n) | O(n^2) | O(n^2) |
| Сортировка вставками | Массив | O(n) | O(n^2) | O(n^2) |
| Сортировка выбором | Массив | O(n^2) | O(n^2) | O(n^2) |
| Блочная сортировка | Массив | O(n+k) | O(n+k) | O(n^2) |
| Поразрядная сортировка | Массив | O(nk) | O(nk) | O(nk) |

## Структуры данных

| Структура данных | Временная сложность | | | | | | |
|---|---|---|---|---|---|---|---|
| | В среднем | | | | В худшем | | |
| | Индексация | Поиск | Вставка | Удаление | Индексация | Поиск | Вставка |
| Обычный массив | O(1) | O(n) | - | - | O(1) | O(n) | - |
| Динамический массив | O(1) | O(n) | O(n) | O(n) | O(1) | O(n) | O(n) |
| Односвязный список | O(n) | O(n) | O(1) | O(1) | O(n) | O(n) | O(1) |
| Двусвязный список | O(n) | O(n) | O(1) | O(1) | O(n) | O(n) | O(1) |
| Список с пропусками | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(n) | O(n) | O(n) |
| Хеш таблица | - | O(1) | O(1) | O(1) | - | O(n) | O(n) |
| Бинарное дерево поиска | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(n) | O(n) | O(n) |
| Декартово дерево | - | O(log(n)) | O(log(n)) | O(log(n)) | - | O(n) | O(n) |
| Б-дерево | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n |
| Красно-черное дерево | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n |
| Расширяющееся дерево | - | O(log(n)) | O(log(n)) | O(log(n)) | - | O(log(n)) | O(log(n |
| АВЛ-дерево | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n |

## Кучи

| Куча | Временная сложность | | | |
|---|---|---|---|---|
| | Преобразование к куче | Поиск максимума | Извлечение максимума | Увеличить ключ |
| Связный список (отсортированный) | - | O(1) | O(1) | O(n) |
| Связный список (не отсортированный) | - | O(n) | O(n) | O(1) |
| Бинарная куча | O(n) | O(1) | O(log(n)) | O(log(n)) |
| Биномиальная куча | - | O(log(n)) | O(log(n)) | O(log(n)) |
| Фибоначчева куча | - | O(1) | O(log(n)) | O(1)* |

## Представление графов

Пусть дан граф с |V| вершинами и |E| ребрами, тогда

| Способ представления | Память | Добавление вершины | Добавление ребра | Удаление вершины |
|---|---|---|---|---|
| Список смежности | O(\|E\| + \|V\|) | O(1) | O(1) | O(\|E\| + \|V\|) |
| Список инцидентности | O(\|E\| + \|V\|) | O(1) | O(1) | O(\|E\|) |
| Матрица смежности | O(\|V\|^2) | O(\|V\|^2) | O(1) | O(\|V\|^2) |
| Матрица инцидентности | O(\|V\|\|E\|) | O(\|V\|\|E\|) | O(\|V\|\|E\|) | O(\|V\|\|E\|) |

## Нотация асимптотического роста

| Обозначение | Граница | Рост |
|---|---|---|
| (Тета) Θ | Нижняя и верхняя границы, точная оценка | Равно |
| (O - большое) O | Верхняя граница, точная оценка неизвестна | Меньше или равно |
| (o - малое) o | Верхняя граница, не точная оценка | Меньше |
| (Омега - большое) Ω | Нижняя граница, точная оценка неизвестна | Больше или равно |
| (Омега - малое) ω | Нижняя граница, не точная оценка | Больше |

1. (O — большое) — верхняя граница, в то время как (Омега — большое) — нижняя границ
так и (Омега — большое), поэтому она является точной оценкой (она должна быть огран
примеру, алгоритм требующий Ω (n logn) требует не менее n logn времени, но верхняя г

требующий Θ (n logn) предпочтительнее потому, что он требует не менее n logn (Ω (n log logn)).

2. f(x)=Θ(g(n)) означает, что f растет так же как и g когда n стремится к бесконечности. Друг асимптотически пропорциональна скорости роста g(n).

3. f(x)=O(g(n)). Здесь темпы роста не быстрее, чем g (n). O большое является наиболее поле наихудший случай.

Короче говоря, если алгоритм имеет сложность __ тогда его эффективность __

| Алгоритм | Эффективность |
|---|---|
| o(n) | < n |
| O(n) | ≤ n |
| Θ(n) | = n |
| Ω(n) | ≥ n |
| ω(n) | > n |

## График роста O — большое