SkillFactory

Fullstack разработчик на Python

Модуль: Алгоритмы и структуры данных

Сложность алгоритма

“O” большое	верхняя граница сложности
Θ (Тета)	точная оценка сложности
Ω (Омега)	нижняя граница сложности

Сложность алгоритма

O(1)	константная
O(log(n))	логарифмическая
O(n)	линейная
O(n log(n))	линейно-логарифмическая
O(n^k)	полиномиальная
O(n!)	факториальная

Операции с массивами

получение размера	**O(1)**
элемент по индексу	**O(1)**
вставка в конец/ удаление с конца	**O(1)**
вставка в конец (с расширением)	**O(n)**
вставка/удаление (с произвольного места)	**O(n)**

Хэш-таблицы

вставка пары	**O(1)**
удаление пары	**O(1)**
получение значения по ключу	**O(1)**
получение размера	**O(n)**

Односвязный список

вставка в конец/начало	O(1)
вставка на произвольное место	O(1)
удаление из начала	O(1)
удаление с произвольного места	O(n)
получение размера списка	O(n)

Стек

вставка наверх (push)	**O(1)**
удаление сверху (pop)	**O(1)**
верхний элемент (top)	**O(1)**
размер (size)	**O(1)**

Очередь на циклическом массиве

вставка в конец	**O(1)**
удаление из начала	**O(1)**
получение первого элемента	**O(1)**
общий размер	**O(1)**

Графы

наличие ребра	**O(|E|)**	**O(1)**
степень вершины	**O(1)**	**O(|V|)**
память	**O(|V|+|E|)**	**O(|V|^2)**
вставка/ удаление	**O(1)**	**O(d)**
обход графа	**O(|V|+|E|)**	**O(|V|^2)**
алгоритм Дейкстры (время)	**O(|V|^2+|E|)**	
алгоритм Дейкстры (доп.память)	**O(|V|)**	

|V| - количество вершин
|E| - количество ребер
d - степень вершины

SkillFactory

Fullstack разработчик на Python

Модуль: Алгоритмы и структуры данных

Алгоритм Дейкстры

Цель: поиск минимального пути в графе
1. Создание словаря расстояний D
2. Создание словаря просмотренных вершин U
3. Выбор вершины с наименьшим d
4. Во все непросмотренные вершины записать расстояния до них
5. Повтор п.3-4 |V| раз

Бинарные деревья

Бинарное дерево - это граф, обладающий свойствами:
1. связный (нет вершин без ребер)
2. не имеет циклов
3. неориентированный
4. невзвешенный
5. имеет не более 2 потомков

Виды обхода деревьев

обход в глубину (DFS)	префиксный (**pre-order**)
	инфиксный (**in-order**)
	постфиксный (**post-order**)
обход в ширину (BFS)	

Алгоритмы поиска

линейный поиск	**O(n)**
двоичный поиск	**O(log(n))**
поиск в графе	**O(n)**
двоичное дерево поиска	**O(log(n))**
поиск в хэш-таблице	**O(1)**

Сортировка пузырьком

“всплывание” максимума вправо	
Лучший случай	**O(n)**
Средняя оценка	**O(n^2)**
Худший случай	**O(n^2)**

Сортировка вставками

поиск положения в отсортированной части	
Лучший случай	**O(n)**
Средняя оценка	**O(n^2)**
Худший случай	**O(n^2)**

Сортировка слиянием

“разделяй и властвуй” деление на части и слияние с сортировкой	
Лучший случай	**O(n log(n))**
Средняя оценка	**O(n log(n))**
Худший случай	**O(n log(n))**

Быстрая сортировка

“разделяй и властвуй” деление массива на части относительно опорного элемента	
Лучший случай	**O(n log(n))**
Средняя оценка	**O(n log(n))**
Худший случай	**O(n^2)**