**Дополнительно задание на тему «Простейшие структуры данных»**

Ответьте *РАЗВЕРНУТО*на следующие вопросы. Документ с ответами загрузите в свой репозиторий, после чего создайте коммит с названием «C3 additional complete». Напишите мне в дискорде о выполнении, чтобы получить баллы.

1. Что такое Хэш-таблица? Какие способы реализации Хэш-таблиц вам известны?

Хэш-таблица – это структура данных предназначенная для реализации ассоциативных массивов в которых адресация реализуется с помощью хэш-функций. Хэш-функция – функция, преобразующая ключ в некий индекс h(x), который используется для адресации хеш-функции. Основным преимуществом хэш-функций как структуры хранение данных является тот факт, что операции вставки, поиска и удаления выполняются в хэш-таблицах за минимальное время O(1).

Существует две реализации хэш-таблиц отличающиеся методами решения проблемы коллизии (ситуации при которой индекс h(x) соответствует нескольким различным ключам): цепочкой и смещением.

При реализации хэш-таблицы цепочкой, в таблице хранятся ссылки на соответствующие односвязные списки, в которых содержатся ключи и полезные данные. При добавлении нового элемента в хэш-таблицу, этот элемент ставится в начало соответствующего ему по h(x) списка.

При реализации хэш-таблицы методом смещения, берётся натуральное число k. При попытке вставить элемент в занятую ячейку h(x), проверяется сначала ячейка h(x)+k, потом h(x)+2k и т.д. пока не найдётся пустая ячейка в которую будет помещён новый элемент. При поиске в такой таблице проверяется сначала ячейка h(x)+k, потом h(x)+2k и т.д. пока не найдётся элемент с искомым ключом. При удалении, каждый элемент таблицы с h(x) <= h(x) удаляемого элемента смещается на k ячеек влево в таблице.

2. Что такое граф? Какие способы реализации графа вам известны?

Граф – совокупность вершин и рёбер, соединяющих вершины. Путь в графе – множество рёбер, ведущих из одной вершины в другую. Цикл – путь, начинающийся и заканчивающийся в одной и той же вершине. Связный граф – граф, в котором существует путь из любой вершины в любую другую. Компонент связности некого графа – максимальный по включение связный подграф некого графа. Ориентированный граф – граф в котором рёбра имеют направление. Взвешенный граф – графдля которого определена функция ω(E). Степень вершины – число верши с которыми связана данная вершина.

Способы реализации графа: список рёбер, матрица смещения, списки смещения.

Список рёбер – список, каждый элемент которого имеет три значения: номер массива из которого ребро выходит, номер массива в который оно заходит и вес ребра. Данный способ реализации графа не способен отображать рёбра со степенью 0, а также малопригоден для большинства операций проводимых на графах.

Матрица смещения – матрица, по строкам и столбцам которой расположены номера вершин, а на пересечении соответствующих вершин находится вес ребра, проходящего из одной вершины в другую. Преимуществом является скорость выполнения различных операций за O(1), недостатком – высокий расход памяти.

Списки смещения. Данная реализация представляет из себя массив, каждая ячейка которого соответствует одной вершине и содержит список рёбер начинающихся в данной вершине. Каждый элемент списков в сою очередь содержит номер вершины, в которой ребро кончается и его вес. Данная реализация медленнее матрицы смещения, но расходует меньше памяти.

3. Какие алгоритмы на графах вам известны? Опишите их.

-Обход в глубину:

1. Выбирается произвольная не пройденная вершина.

2. Вершина помечается как пройденная (при этом над ней проделываются необходимые действия).

3. До тех пор пока у вершины есть не пройденные выбирается произвольный сосед вершины и к нему применяются шаги 2 и 3.

Если после завершения работы алгоритма в графе остались не

пройденные вершины, то он выполняется ещё раз.

-Обход в ширину

1. Произвольная не пройденная вершина добавляется в очередь.

2. Следующее выполняется пока очередь не пуста:

2.1 Берётся следующий элемент из очереди.

2. 2 Элемент отмечается как посещённый.

2.3 В очередь помещаются все ещё не помещённые и не посещённые соседи.

Если после завершения работы алгоритма в графе остались не

пройденные вершины, то он выполняется ещё раз.

-Подсчёт компонентов связности:

Проводится обход в глубину, число выполнений 1-го шага обхода равно числу компонентов связности графа.

-Подсчёт кротчайших путей в невзвешенных графах:

Производится обход в ширину, при этом вместо произвольной вершины в 1-ом шаге берётся вершина путь из которой ищется. Этой вершине присваивается 0. Каждый раз когда в ходе обхода сосед текущего элемента помещается в очередь ему присваивается число равное числу текущего элемента +1. Число присвоенное каждой вершине –кратчайший путь до этой вершины из изначальной вершины.

-Подсчёт кротчайших путей в взвешенных графах:

Вершине путь из которой ищется присваивается 0, остальным вершинам присваивается ∞. Выполняется следующий алгоритм:

1. Берётся элемент с наименьшим присвоенным значением.

2. Для каждого соседа элемента выполняется:

2.1 Значение элемента суммируется с весом ребра ведущего к соседу.

2.2 Сумма сравнивается со значением соседа. Если она меньше, то соседу присваивается новое значение равное сумме.

3.Элемент отмечается пройденным

Этот алгоритм продолжает выполняется до тех пор пока существуют непройденные элементы-соседи пройденных элементов.

После этого число присвоенное каждой вершине – вес кратчайшего пути до этой вершины из изначальной вершины.