Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**КУРСОВОЙ ПРОЕКТ**

Реализация Skip List

по дисциплине: «Алгоритмы и структуры данных»

Выполнил студент гр. 3530901/80202 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.А. Иванов

(подпись)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.И Глухих

(подпись)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

.

Санкт-Петербург

2020

**ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА**

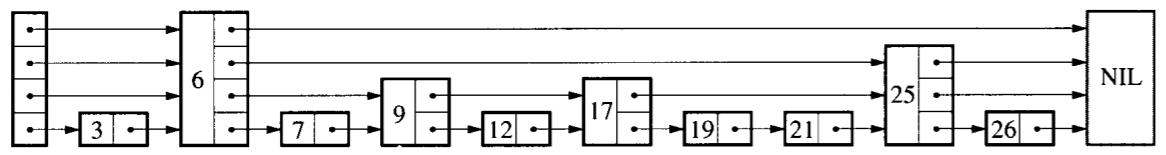
1. Тема проекта(работы): Написать класс для реализации Skip List
2. Срок сдачи законченной работы: 10 февраля
3. Исходные данные к работе: Язык программирования Kotlin, cреда разработки IntelliJ IDEA.
4. Содержание работы: Список с пропусками, с примером использования в приложении.

**ОПИСАНИЕ ЗАДАНИЯ И РЕАЛИЗАЦИИ**

Список с пропусками – это вероятностная структура данных, позволяющая в среднем за O(log(n)) (в худшем за O(n)) времени выполнять операции поиска, добавления и удаления элементов. Эта структура данных может быть реализована разными способами, будь это несколько, параллельных, отсортированных связных списков, или же просто хранение слоев, конечно есть и другие виды реализаций (Sequential, Lock-Free, Fine-Grained Skip List’s) в которых тоже могут варьироваться различные виды внутренних значений (например, вероятность добавления нового элемента на более высокий уровень, при добавлении нового элемента).

В основе списка с пропусками лежит расширение отсортированного связного списка дополнительными связями, добавленными в случайных путях с геометрических распределением, таким образом, чтобы поиск по списку мог быстро пропускать части этого списка.

В своей реализации я выбрал такой вид списка с пропусками, о котором говориться, в указанном списке литературы (Pugh W.) - как уже было сказано выше, состоит из нескольких уровней (слоев), на каждом из которых расположен указатель на не на следующий элемент, а через некоторое количество элементов и чем длиннее, а соответственно и выше список, то если мы будет подниматься по этим слоям как по лестнице, то будем обнаруживать все меньшее количество указателей, следующих через большее расстояние нежели их соседи в нижнем слое. На самом нижнем (или первом) уровне располагаются все элементы списка, также в упорядоченном виде.

s

Основные операции в структуре (которые и были мной реализованы), выполняемые за логарифмическое случайное время:

* Поиск элемента
* Вставка элемента
* Удаление элемента

Конечно, в некоторых случаях, будь то нахождение, добавление или удаление элемента, мы можем получить плохое время выполнения, но конечно же такие выполнения достаточно редки. Список с пропусками имеет маловероятную возможность быть “плохой” структурой данных в конкретном случае, хоть он и имеет плохую производительность (big O) в худшем случае, никакая входная последовательность не дает стабильно худшей производительности. Ведь список с пропусками и представление в нем данных всего лишь связанные списки с дополнительными указателями, которые могут ‘пролетать’ над промежуточными узлами.

**Список использованной литературы:**

1. Pugh, W. (1990). Skip lists: a probabilistic alternative to balanced trees. Communications of the ACM, v.33(n.6), pages 668–676. doi:10.1145/78973.78977
2. Список с пропусками - https://neerc.ifmo.ru/wiki