Київський національний університет ім. Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Група К-28

Пономарьов Микита Максимович

**Лабораторна робота №8**

м. Київ, 2020р

**Умова**: Реалізація піраміди Фібоначчі.

**Реалізація**:Купа Фібоначчі являє собою набір дерев, кожне з яких є мінкупою змінної арності з такими властивостями:

* Вузли дерев відповідають елементам, що зберігаються в черзі.
* Корені куповпорядкованих дерев поєднані у двобічно зв'язаний список.
* Зберігається вказівник на корінь дерева, який відповідає елементу з найменшим ключем (варто зауважити, що те, що кожне дерево є мінкупою, гарантує, що цей елемент буде коренем одного з дерев).
* Для кожного вузла зберігається його ранг (степінь), тобто кількість його дітей, а також чи він позначений (мету позначення ми визначимо пізніше).
* *Вимога розміру*: якщо вузол *u* має степінь *k*, то піддерево з коренем *u* має щонайменше вузлів.

Кожен вузол *x* містить вказівник *x.p* на свого предка і вказівник *x.child* на один із його нащадків. Нащадки зв'язані в циклічний [двобічно зв'язаний список](https://uk.wikipedia.org/wiki/Двобічно_зв'язаний_список). Кожен нащадок *y* має вказівники *y.left* та *y.right.* Якщо вузол є єдиним нащадком, тоді *y = y.left = y.right.* Нащадки елемента можуть перебувати у будь-якому порядку. Використання двобічно зв'язаних списків уможливлює вставляння і видалення елементів за час *O(1),* а також зв'язування двох списків за *O(1).* Також кожен вузол має атрибут *x.degree,* що дорівнює кількості нащадків, і атрибут *x.mark,* що показує, чи втратив вузол нащадка з моменту, як він став нащадком свого поточного предка. Піраміда підтримує наступні операції:

* **Вставка**: додаємо новий елемент *s* як нову мінкупу в колекцію і перевіряємо, чи *k(s)* менше за поточне найменше значення. Якщо так, то відповідно змінюємо вказівник *H.min*.
* **Зменшення ключа**: Коли ми зменшуємо ключ елемента *s*, якщо умови мінкупи все ще задовільняються, то нам більше не потрібно робити нічого. Інакше, ми просто відтинаємо піддерево з коренем в *s* і вставляємо його як нове піддерево у колекцію. Ми порівнюємо новий ключ *s* із попереднім мінімальним елементом і змінюємо вказівник *H.min* відповідно.
* **Отримання мінімуму**: Ми можемо описати видобування мінімального елемента *s\**. Починаємо з видалення *s\** (на нього вказує *H.min*) і додавання усіх його дітей як дерев у колекцію. Тепер, ми продивляємось усю колекцію, знаходимо найменший елемент і оновлюємо *H.min* відповідно.