**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

**Кафедра інформаційних систем**

**Алгоритми та складність**

**Лабораторна робота № 3**

**Звіт**

**Виконав:**

студент групи К-28

Макарчук Іван

**Київ-2018**

**Зміст завдання**

Представити d-арну піраміду у вигляді масиву (висота піраміди log(d,n), де d – основа, n – кількість елементів масиву);

Піраміда – частково упорядкована структура даних, яку можна розглядати як d-арне дерево з певними властивостями:

•Дерево заповнене на всіх рівнях крім, можливо, останнього.

• Останній рівень заповнюється зліва направо до вичерпання елементів.

• Ключі у вузлах певним чином упорядковані.

• d-арнапіраміда може бути ефективно реалізована у вигляді масиву шляхом запису її елементів згори донизу зліва направо.

• Висота d-арної пірамідидля n елементів дорівнює

**Основні модулі програми**

* intParent(int i)

Визначає індекс батька і-го вузла

Батько і-го вузла буде знаходитись за формулою (i - 1) / d;

Оскільки для d-ого сина вузла parent, його індекс буде d\*(parent+1)(тому що нумерація починається з 1, а не з 0), і тому батько i-го вузла буде знаходитись за формулою

(i - 1) / d;

* intSon(int parent, int k)

Визначає індекс k-го сина у вузлі, батьком якого є parent

На вхід приймається масив А, індекс вузла і та номер k його сину.

Усі сини вузла і мають індекси від d\* parent +1 до d\* parent + parent

Відповідно k-ий син вузла і має індекс d\* parent +k;

* void Max\_Heapify(vector<int>&A, inti)

**Час роботи**: О(d\*)

Підтримує властивість незростання піраміди;

На вхід приймає масив А та індекс вузла і;

Вважається, що дерева з коренями Son(i, k) (k =1-d) –

незростаючі піраміди, але елемент A[i] може порушувати

цю властивість.Значення A[i] просувається вниз, поки відповідне дерево з

цим коренем не стане незростаючою пірамідою.

* intHeap\_Extract\_Max(vector<int>&A)

**Час роботи**: О(d\*)

На вхід приймає масив А;

Повертає і вилучає максимальне значення в піраміді;

За побудовою піраміди максимальне значення знаходиться у корені, воно вилучається, на його місце ставиться останній елемент масива і викликається процедура Max\_Heapify(A, 0)

* voidHeap\_Increase\_Key(vector<int>&A, int i, intkey)

**Час роботи**: О()

На вхід приймає масив А, позицію і в піраміді та ключ key;

Змінює ключ елемента з індексом і на новий ключ key(за умови що значення нового ключа не менше за старе);

Після заміни ключа йде перевірка, чи не порушилась властивість незростання. В такому випадку відбувається обмін з батьківським елементом, поки властивість не відновиться.

* voidMax\_Heap\_Insert(vector<int>&A, intkey)

**Час роботи**: О()

На вхід приймає масив А та ключ key;

Вставляє в піраміду новий вузол зі значенням key;

**Література :**

<https://www.geeksforgeeks.org>

https://stackoverflow.com/