Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інформаційних систем

Лабораторна робота №3

“D-арні піраміди”

Виконав студент 2го курсу

Групи К-29

Печкурова Віталія Михайловича

2018

**Завдання**: представити d-арну піраміду у вигляді масиву, реалізувати процедури Heap\_Extract\_Max, Max\_Heap\_Insert та Heap\_Increase\_Key.

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-арна піраміда може бути ефективно реалізована у вигляді масиву шляхом запису її елементів згори донизу зліва направо.

• Висота d-арної піраміди для n елементів дорівнює

**Модулі програми:**

* int Parent (int i)

Визначає індекс батька і-го вузла

* int Son(int parent, int k)

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

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

**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.

* int Heap\_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)

* void Heap\_Increase\_Key(vector<int> &A, int i, int key)

**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.

* void Max\_Heap\_Insert(vector<int> &A, int key)

**Час роботи**: О()

На вхід приймає масив А та ключ key;

Вставляє в піраміду новий вузол зі значенням key;

**Тестові приклади:**

* d = 4

A = {10, 29, 28, 27, 26, 25, 20, 19, 17, 15}

Збільшення ключа: і = 5, key = 40

Вставка елемента: key = 13

Графічне зображення роботи Max\_Heapify(A, 0)

* d = 3 A = {36, 80, 41, 72, 28,10, 15, 40, 37, 11, 10, 18, 63, 13}

Збільшення ключа: і = 6, key = 42

Вставка елемента: key = 31

**Література:**

* Лекція №5
* <http://study.sfu-kras.ru/DATA/docs/ProgramTheory/recurs/s_sort.htm>