Київський національний університет імені Тараса Шевченка

Факультет комп'ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Завдання №7

“Біноміальна піраміда”

Виконав студент 2-го курсу

Групи К-29

Демиденко Андрій Геннадійович

2021

**Предметна область**

Предметна область: Відділ кадрів

Об’єкти: Підрозділи, Співробітники

Примітка: Існує множина підрозділів. Кожен підрозділ включає в себе множину співробітників.

**Завдання**

Реалізувати біноміальну піраміду. У вузлах піраміди будуть зберігатись інформація про підрозділи.

**Теорія**

Біноміальна піраміда (біноміальна купа) Н – множина біноміальних дерев, що задовольняють властивостям біноміальних пірамід:

* Кожне біноміальне дерево в Н є неспадаючою пірамідою (мінімальний елемент на вершині);
* Для довільного невід’ємного k в Н існує не більше одного біноміального дерева відповідного порядку.
* Біноміальна піраміда з n вузлів складається не більше ніж з (lg 𝑛 + 1) біноміальних дерев.

**Алгоритм**

**Insert**

Створюємо піраміду з одного елементу і об’єднуємо (Merge)

**ExtractMin**

Видаляємо мінімум, який знаходиться в кореневому списку, тобто на найнижчому. Об'єднуємо вихідну піраміду з перевернутим списком його дітей.

**Merge**

1. Спочатку об'єднаємо кореневі списки куп в один кореневий список, підтримуючи впорядкованість по degree.
2. Зберігаємо покажчики на початок списків і в результуючий список записуємо мінімальний з них, той звідки щойно записали зрушуємо на наступний.
3. Далі проходимся від початку до кінця нового отриманого кореневого списку і зливаємо дерева однакового розміру в одне. Можуть бути випадки:

1) Тільки 2 дерева однакового розміру. Тоді об'єднуємо їх.

2) 3 дерева однакового розміру. Об'єднуємо 2 останніх.

При об'єднанні двох дерев потрібно лише подивитися, в корені якого з них менший ключ і зробити інше дерево лівим сином кореня цього дерева.

**Складність**

* Час пошуку мінімального значення O(lg n), бо перевіряємо не більше

коренів;

* Час злиття двох біноміальних пірамід Ο(lg n);
* Вставка вузла. Створюється біноміальна піраміда з одним вузлом

(час Ο(1)) та зливається з початковою пірамідою (загальний час Ο(lg n));

* Час вилучення мінімального вузла Ο(lg n).

**Мова програмування**

С++

**Основні модулі програми**

//Class for implementation heap node

template <typename T>

class Node

{

private:

Node\* parent;

Node\* sibling;

Node\* child;

T value;

int degree{};

public:

//Constructors

Node();

explicit Node(T newValue);

//Connection between nodes

void Link(Node\* other)

//Outputting heap in console

void Print(int countTabs) const

friend BinominalHeap<T>;

};

//Class for implementation binominal heap

template <typename T>

class BinominalHeap {

private:

Node<T>\* head;

//Some helpful functions

public:

//Constructors

BinominalHeap()

explicit BinominalHeap(T newValue)

//Destructor

~BinominalHeap() = default;

//Function for outputting heap

void Print() const

//Insertion new element

void Insert(const T& value)

//Function for deletion the smallest element from heap

void ExtractMin()

//Function for decreasing key

void DecreaseKey(Node<T>\* node, T newValue)

//Function for deletion any node

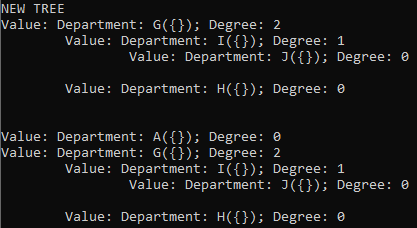
void Erase(Node<T>\* node)

};

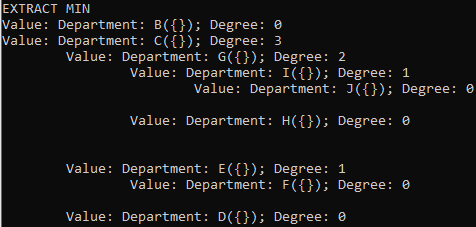
**Інтерфейс користувача**

Вхідні дані генеруються програмою і виводяться на консоль. За потреби можливо налаштувати програму на ввід з консолі.

**Приклад виводу програми**

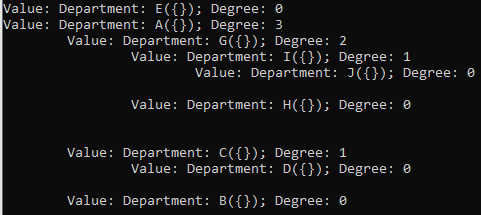
****

Видалення найменшого:

****

**Тестові приклади**

Нехай у нас є наступна піраміда:



Перемалюємо її у звичному для нас вигляді:

E

A

C

B

G

I

H

D

J

Додавання вузла F:

E

A

C

B

F

G

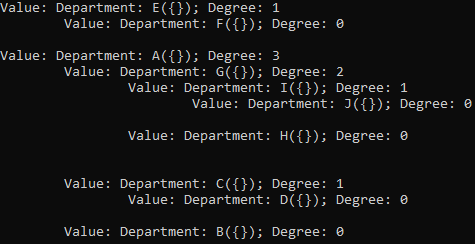
I

H

D

J

Вивід програми:



Видалення найменшого вузла (А)

Видаляється вузол A, а потім в оберненому порядку додаються всі його ключі

Початкове дерево:

E

A

B

C

F

G

I

H

D

J

Після видалення:

B

C

E

D

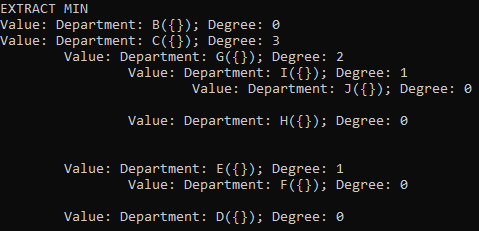
G

I

F

J

Вивід програми



**Висновки**

В практичних задачах вхідні дані формуються випадковим чином. Малоймовірно, щоб вони розташувались у строгому порядку. Тому при запису в піраміду складність, як правило, буде *O(*log*n).*

**Література**

* Лекція №5
* “Кучи. Часть 1. Биномиальная куча” (https://habr.com/ru/post/135232/)
* “Биномиальные пирамиды” (http://www.myshared.ru/slide/1303746/)