

Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем
Алгоритми та складність

Завдання №7
“Біноміальна піраміда”
Виконав студент 2-го курсу
Групи К-28
Гуща Дмитро Сергійович

Предметна область

Варіант 4

Предметна область: Учбовий відділ

Об'єкти: Групи, Студенти

Завдання

Реалізувати біноміальну піраміду для збереження об'єктів з предметної області.

Теорія

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

1. Кожне біноміальне дерево в H є неспадаючою пірамідою (мінімальний елемент на вершині);
2. Для довільного невід'ємного k в H 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 в кореневому списку. Об'єднуємо вихідну піраміду з перевернутим списком його дітей.

(Merge)

Спочатку об'єднаємо кореневі списки куп в один кореневий список, підтримуючи впорядкованість по degree.

Зберігаємо покажчики на початок списків і в результуючий список записуємо мінімальний з них, той звідки щойно записали зрушуємо на наступний.

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 $(\lg n + 1)$ коренів
- Час злиття двох біноміальних пірамід $O(\lg n)$
- Вставка вузла. Створюється біноміальна піраміда з одним вузлом (час $O(1)$) та зливається з початковою пірамідою (загальний час $O(\lg n)$)
- Час вилучення мінімального вузла $O(\lg n)$

Мова програмування

C++

Основні модулі програми

student.h

```
class Student{}; //Клас опису студента
std::string getName(); // метод повертає ім'я студента
void getStudent(); //метод виводить ID та ім'я студента в консоль
void setName(std::string name); //метод змінює ім'я студента
```

group.h

```
class Group {};
```

```
Group() : title("NULL"); //конструктор пустої групи
Group(std::string title); //конструктор з початковою назвою групи
Group(std::string title, Student* first_student); //конструктор з початковою назвою
групи та першим студентом
std::string getGroupTitle(); //модуль повертає назву групи
std::vector<Student*> getGroupStudents(); //модуль повертає множину студентів
void setGroupTitle(std::string title); //модуль змінює назву групи
void setGroupStudents(std::vector<Student*> students); //модуль змінює множину
студентів
void addStudent(Student* student); //додати нового студента
void printStudents(); //вивід у консоль усіх студентів групи
```

binHeap.h

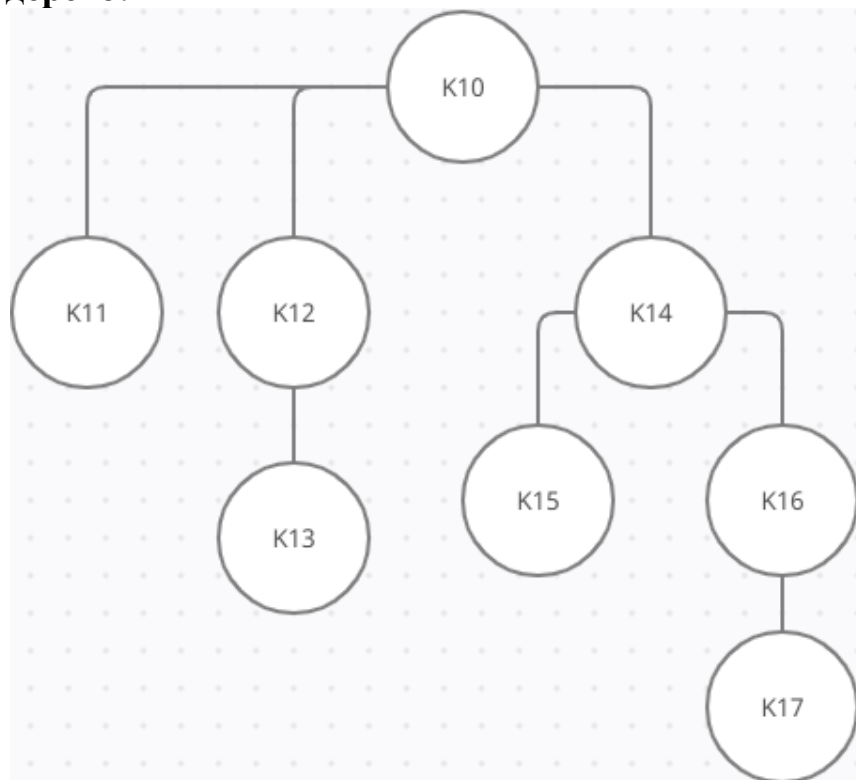
```
class Node; //клас для опису вузла біноміальної піраміди
void print(int countTabs) const; //функція виводу вузла
void link(Node* other); //вставка поточного вузла у дочірній вузол вузла
other
class BinominalHeap; //клас для опису біноміальні піраміди
void merge(BinominalHeap<T>* first, BinominalHeap<T>* second); //функція
злиття двох біноміальних пірамід
void unionHeaps(BinominalHeap<T>* first, BinominalHeap<T>* second); //виконує
операцію unionheap
void insert(T value); //функція вставки вузла у піраміду
void extractMin(); //функція витягнення мінімального елементу
void decreaseKey(Node<T>* node, T newValue); //функція збільшення ключа
[[nodiscard]] Node<T>* minNode() const; //пошук мінімального елементу
[[nodiscard]] Node<T>* prevMinNode() const; //пошук переднайменшого елементу
```

Інтерфейс користувача

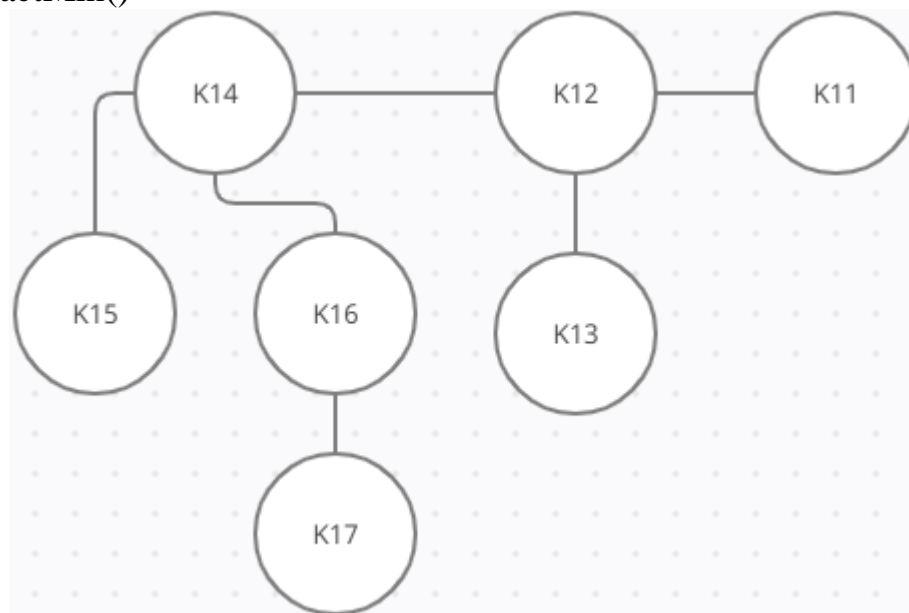
Вхідні дані задаються автоматично при запуску програми і результат виводиться в консоль

Тестовий приклад:

Початкове дерево:



Після extractMin()



Висновки

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

Література

- Лекція №5
- <https://habr.com/ru/post/135232/>
- <http://www.myshared.ru/slide/1303746/>