Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Завдання №7

“Реалізація біноміальної піраміди”

Виконав студент 2-го курсу

Групи К-29

Циганов Аркадій Олексійович

2021

**Предметна область**

Предметная область  Кафедра университета

Объекты  Преподаватели, Дисциплины

Примечание  На  кафедре  имеется    множество преподавателей.

Для  каждого преподавателя  задано  множество дисциплин.

**Завдання**

Реалізувати біноміальноу піраміду

**Теорія**

**Біноміальна купа** ([англ.](https://uk.wikipedia.org/wiki/Англійська_мова) *binomial heap*) — це множина [біноміальних дерев](https://uk.wikipedia.org/wiki/Біноміальне_дерево), що задовольняє властивостям біноміальної купи:

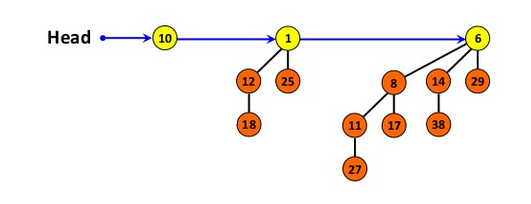
1. Кожне біноміальне дерево у купі підпорядковується *властивості неспадної купи* ([англ.](https://uk.wikipedia.org/wiki/Англійська_мова) *min-heap property*): ключ вузла не менший за ключ його батьківського вузла.
2. Для будь-якого невід'ємного цілого *k* в купі існує не більше одного біноміального дерева, чий корінь має ступінь *k*.

З даних властивостей випливає, що біноміальна купа, що має *n* вузлів, складається з не більше ніж біноміальних дерев.

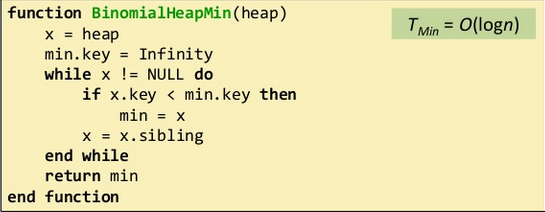
Завдяки своїй структурі, біноміальне дерево ступеня *k* можна побудувати з двох дерев ступеня *k*−1 тривіальним приєднанням одного з них до іншого як найлівішого підпорядкованого дерева. Ця властивість є центральною для операції *злиття* біноміальних дерев, яка становить їхню основну перевагу над звичайними купами.

Ім'я походить від того факту, що біноміальне дерево ступеня  має  вузлів на глибині .

**Алгоритм**



**Пошук мінімальногоузла (FindMin)**



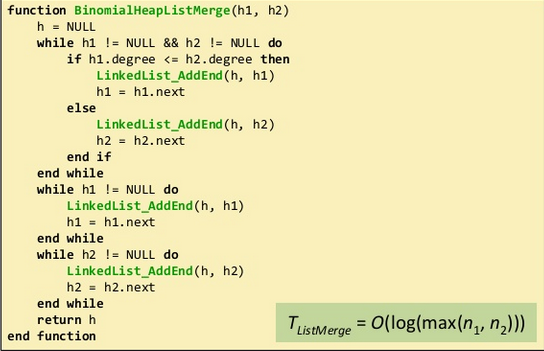
**Злиття куп (Union)**

Для злиття (union, merge) двох куп H1 і H2 в нову купу необхідно:

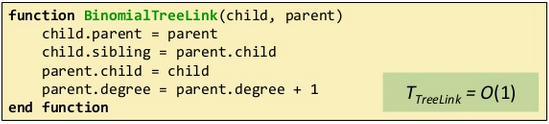
1.Сліть списки коренів h1и H2в один впорядкований список

2.Восстановіть властивості біноміальної купи H 

Після злиття списків коренів відомо, що в купі Hімеется не більше двох коренів з однаковим ступенем і вони є сусідами

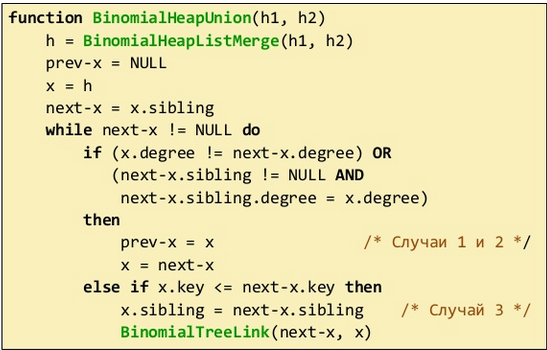


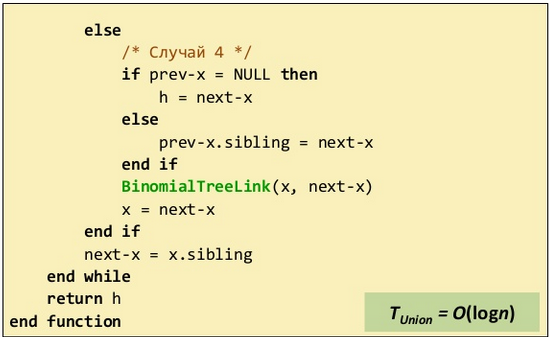
**Зв'язування біноміальних дерев**



**Злиття біноміальних куп (Union)**

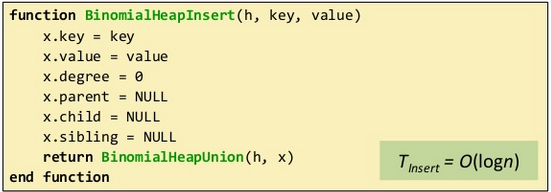
Обчислювальна складність злиття двох біноміальних куп в гіршому випадку дорівнює O (log (n)) Довжина списку коренів не перевищує log (n1) + log (n1) + 2 = O (log (n)) Цикл while в функції BinomialHeapUnion виконується не більше O (log (n)) раз На кожної ітерації циклу покажчик переміщається по списку коренів вправо на одну позицію або віддаляється один вузол -це вимагає часу O (1)





**Вставка вузла (Insert)**

Створюємо біноміальними купу з одного вузла x- біноміального дерева B0 зливаємо ну купу H і купу з вузла x.



**Видалення мінімального вузла (DeleteMin)**

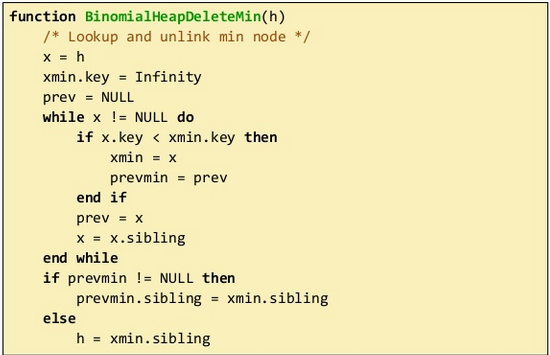
1.В списку коренів купи H шукаємо корінь x з мінімальним ключем і видаляємо x із списку коренів (розриваємо зв'язок)

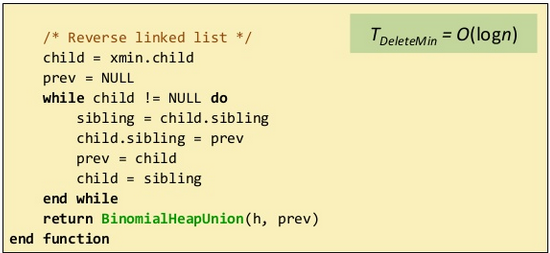
2.Ініціалізуєм порожню купу Z

3.Змінюємо порядок проходження дочірніх вузлів кореня х на зворотний, у кожного дочірнього вузла встановлюємо поле parent в NULL

4. Встановлюємо заголовок купи Z на перший елемент нового списку коренів 5.Зливаємо купи Hи Z

6.Повертаємо x





**Складність**

**Поиск минимальногоузла(FindMin)** {\displaystyle O(\log(n)).}O

Пошук мінімального вузла (FindMin) - O(log(n))

Злиття куп (Union) n) - O(log(max(n1, n2))

Зв'язування біноміальних дерев - O(1)

Злиття біноміальних куп (Union) - O(log(n))

Видалення мінімального вузла (DeleteMin) - O(log(n))

Вставка вузла (Insert) - O(log(n))

**Мова програмування**

С++

**Модулі програми**

* template <typename T> class BinominalHeap

Головний класс купи, в якому реалізовані функції:

1. void merge (BinominalHeap<T> \*first, BinominalHeap<T> \*second)

2. void unionHeaps (BinominalHeap<T> \*first, BinominalHeap<T> \*second)

3. void insert (T value)

4. void extractMin ()

5. void decreaseKey(Node<T>\* node, T newValue)

6. Node<T>\* minNode()

які викликають відповідні операції для зміни конфігурації купи.

**Інтерфейс користувача**

Вхідні дані задані на етапі запуску програми. На екран виводиться купа після вставки кажного вузла.

**Тестові приклади**

|  |  |
| --- | --- |
| **input** | **output** |
| Порядок вставки предметів із заданою кількістю годин:  0, 5, 6, 4, 3, 7, 9 |  |

**Література**

* Томас Х. Кормен и др. Алгоритмы: построение и анализ. — М. : Видавничий дім «Вільямс», 2007. — 1296 с. — ISBN 5-8459-0857-4.
* Daniel Sleator, Robert Tarjan. A data structure for dynamic trees. — Journal of Computer and System Sciences, 1983. — С. 262-391.