

Київський національний університет імені Тараса Шевченка

Факультет комп'ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Алгоритми та складність

Завдання №3

“ Реалізація біноміальної піраміди для комплексних чисел”

Варіант №4

Виконала студентка 2-го курсу

Групи ІПС-22

Клевчук Марія Вячеславівна

Київ - 2025

Завдання

Реалізувати біноміальну піраміду для типу даних - комплексні числа.

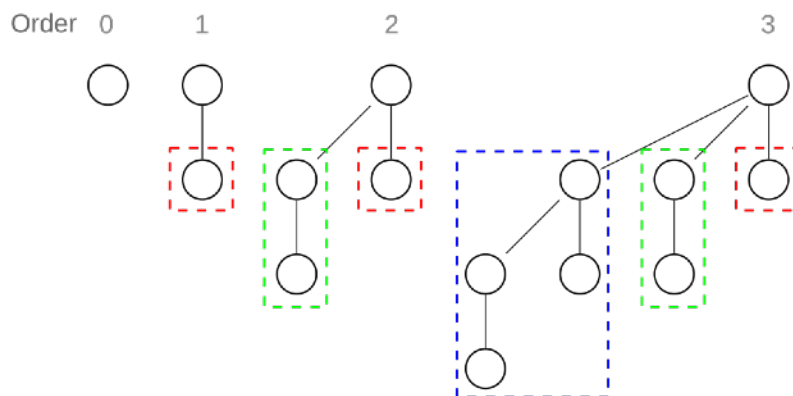
Теорія

Біноміальна піраміда H - множина біноміальних дерев, яка задовольняє таким властивостям біноміальних пірамід:

- кожне біноміальне дерево в H є неспадаючою пірамідою (найменший елемент у вершині дерева);
- в H існує не більше одного біноміального дерева відповідного порядку.

Біноміальна піраміда з n вузлів складається не більше ніж з $(\lceil \lg n \rceil + 1)$ біноміальних дерев.

Біноміальне дерево B_k — це рекурсивно означене упорядковане дерево. Біноміальне дерево B_0 складається з одного вузла. Біноміальне дерево B_k складається з двох біноміальних дерев B_{k-1} з'єднаних разом: корінь одного з них є крайнім лівим дочірнім вузлом кореня другого дерева.



Алгоритм

Операція злиття двох біноміальних пірамід. Спочатку об'єднуємо списки коренів дерев біноміальних пірамід, впорядковуючи їх за порядком.

Після злиття списків коренів відомо, що купа H містить не більше двох коренів однакової степені, і вони стоять підряд. Тому будемо зв'язувати корені однієї степені поки всі корені не отримають різні степені. Можливі два випадки:

- маємо два дерева однакового порядку. Тоді зв'яжемо їх в одне дерево, яке стає на порядок вищим. Дерево з меншим коренем стає батьком, дерево з більшим ключем - його лівим нащадком;
- маємо три дерева однакового порядку. Тоді об'єднуємо два останніх за алгоритмом, описаним вище.

Операція вставка ключа.

Створюється дерево з одним новим ключем. Далі виконується операція злиття двох пірамід - вже існуючої та нової, яка складається з одного елемента.

Операція видалення найменшого ключа.

Шукаємо вузол з мінімальним ключем - це корінь якогось із дерев. Видаляємо його, решта дерева розпадається піддерева. Вони впорядковуються за зростанням порядку, і виконується злиття з рештою піраміди.

Операція зменшення ключа.

Значення ключа у відповідному вузлі змінюється на нове. Далі треба «підняти» цей вузол вгору. Для цього значення його ключа порівнюється з батьківським. Якщо значення нового ключа менше - вони з батьківським міняються місцями.

Складність алгоритму

Час виконання базових операцій (пошук мінімального елемента, злиття двох біноміальних пірамід, вставка вузла, зменшення вузла, видалення вузла) - $O(\lg n)$

Мова реалізації алгоритму

C++

Модулі програми

struct Complex - реалізовує комплексні числа:

- `double modulus() const` — повертає модуль числа;
- `bool operator<(const Complex& other) const` — порівняння за модулем, а якщо рівні — за дійсною частиною;
- `bool operator==(const Complex& other) const` — перевірка на рівність;
- `friend std::ostream& operator<<(std::ostream&, const Complex&)` — вивід у форматі `a+bi`.

struct BinomialNode - реалізація вузла біноміальної піраміди.

class BinomialHeap - клас для реалізації біноміальної купи:

- `void insert(const Complex& key)` — вставляє новий вузол у купу;
- `Complex findMin() const` — повертає мінімальне значення в купі;
- `void deleteMin()` — видаляє вузол з мінімальним ключем;
- `void decreaseKey(BinomialNode* node, const Complex& newKey)` — зменшує значення ключа вузла;
- `void print() const` — виводить структуру купи з назвами дерев `B<ступінь>`.
- `static BinomialNode* merge(BinomialNode* h1, BinomialNode* h2)` — об'єднує два списки дерев.
- `static BinomialNode* unionTrees(BinomialNode* head)` — об'єднує дерева однакового степеня.

- `static void linkTrees(BinomialNode* root, BinomialNode* child)` — робить одне дерево піддеревом іншого.
- `void printTree(BinomialNode* node) const` — виводить всі дерева з позначенням степеня.

Інтерфейс користувача

Користувач вводить кількість ключів у початковій піраміді, та значення цих ключів. У консоль виводиться біноміальна піраміда.

Користувач має змогу обрати операцію через консоль (вставка ключа, видалення ключа з найменшим значенням, зменшення значення ключа). Програма виконує обрану операцію та нова біноміальна піраміда виводиться в консоль.

Тестові приклади

Приклад 1. Введення ключів у консоль, створення піраміди.

$$\begin{array}{ccccc}
 5+1i & \text{-----} & & & 1+2i \\
 | & & & / & | \\
 -2+5i & & & -9-2i & -4+7i \\
 & & & | & \\
 & & & 11+3i &
 \end{array}$$

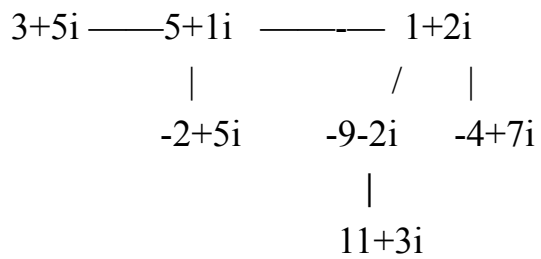
```

Enter number of complex numbers: 6
Enter complex numbers (real imag):
1 2
-4 7
-9 -2
11 3
-2 5
5 1

Initial heap:
B1:
5+1i
  -2+5i
B2:
1+2i
  -9-2i
    11+3i
    -4+7i

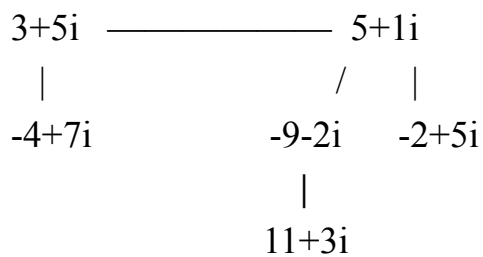
```

Приклад 2. Вставка нового ключа.



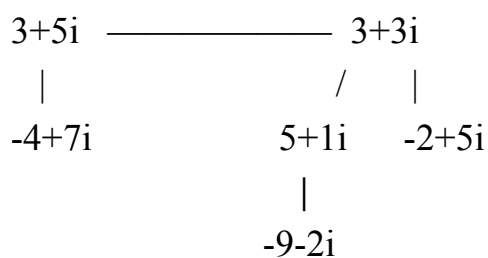
```
Enter complex number to insert: 3 5
B0:
3+5i
B1:
5+1i
-2+5i
B2:
1+2i
-9-2i
11+3i
-4+7i
```

Приклад 3. Видалення найменшого елемента.



```
Menu:
1. Insert
2. Delete Min
3. Decrease Key
0. Exit
Choice: 2
B1:
3+5i
-4+7i
B2:
5+1i
-9-2i
11+3i
-2+5i
```

Приклад 4. Зменшення значення ключа.



```
Enter current complex number (real imag): 11 3
Enter new decreased value (real imag): 3 3
Key decreased successfully.
B1:
3+5i
-4+7i
B2:
3+3i
5+1i
-9-2i
-2+5i
```

Висновки

У результаті виконання лабораторної роботи було реалізовано біноміальні піраміди для комплексних чисел. Основними перевагами цієї структури є компактність зберігання даних, легкий доступ до необхідних елементів та можливість побудови піраміди поетапно, без надмірних обчислень.

Використані літературні джерела

- Алгоритми і складність. Лекція 6
- https://uk.wikipedia.org/wiki/Біноміальне_дерево
- https://uk.wikipedia.org/wiki/Біноміальна_купа
- <https://www.tpointtech.com/binomial-heap>
- <https://web.stanford.edu/class/archive/cs/cs166/cs166.1166/lectures/08/Small08.pdf>