

КОМПЬЮТЕРНОЕ ТЕСТИРОВАНИЕ СТРУКТУРЫ ДАННЫХ, ОСНОВАННОЙ НА ЧИСЛАХ ФИБОНАЧЧИ

С. П. Шаповалов, канд. физ.-мат. наук;

А. А. Стадник, студентка,

Сумский государственный университет, г. Сумы

Реализована структура данных «Фибоначчиева пирамида». Проведено исследование трудоемкости выполнения операций в данной структуре по сравнению с бинарными пирамидами.

Ключевые слова: *структура данных «Фибоначчиева пирамида», трудоемкость выполнения операции.*

Реалізовано структуру даних «Фібоначчієва піраміда». Проведено дослідження складності виконання операцій у даній структурі порівняно з бінарними пірамідами.

Ключові слова: *структура даних «Фібоначчієва піраміда», складність виконання операції.*

ВВЕДЕНИЕ И ПОСТАНОВКА ЗАДАЧИ

Вторая половина XX века ознаменовалась развитием нового научного направления, названного «Теория чисел Фибоначчи» [1,2]. Компьютерные науки подверглись влиянию этой «моды», благодаря чему получен ряд новых результатов в развитии информационных технологий:

- а) разработка класса позиционных представлений чисел – p -кодов Фибоначчи;
- б) разработка систем счисления с иррациональными основаниями;
- в) разработка троичной зеркально-симметричной арифметики;
- г) разработка теории кодирования, основанной на применении теории матриц Фибоначчи;
- д) разработка криптографического метода «Золотая криптография», основанного на применении «золотых» матриц;
- е) разработка структуры данных «Фибоначчиева пирамида».

Внедрение новых идей требует проведения компьютерного тестирования их работоспособности в традиционных алгоритмах по сравнению с уже устоявшимися структурами данных. В данной статье ставится задача сравнения трудоемкости выполнения операций и процедур в фибоначчиевых и бинарных пирамидах.

ФИБОНАЧЧИЕВЫ ПИРАМИДЫ

Фибоначчиева пирамида является реализацией абстрактного типа данных «Очередь с приоритетом». *Фибоначчиева пирамида* – это семейство корневых деревьев, для которого выполняются следующие свойства [3,4]:

- а) каждая вершина в фибоначчиевой пирамиде удовлетворяет основному свойству неубывающей пирамиды;
- б) в семействе корневых деревьев нет двух деревьев с корнями одинакового ранга;
- в) каждая некорневая вершина в фибоначчиевой пирамиде может потерять не более одного «сына» при выполнении процедуры вырезания узла.

СТРУКТУРА ФИБОНАЧЧИЕВЫХ ПИРАМИД

Каждый узел x в фибоначчиевой пирамиде H содержит следующие указатели и поля [2]:

- а) $key[x]$ – поле, в котором хранится ключ;
- б) $p[x]$ – указатель на родительский узел;
- в) $child[x]$ – указатель на один из дочерних узлов;
- г) $left[x]$ – указатель на левый сестринский узел;
- д) $right[x]$ – указатель на правый сестринский узел;
- е) $degree[x]$ – поле, в котором хранится количество дочерних узлов;
- ж) $mark[x]$ – логическое значение, которое указывает, были ли потери узлом x дочерних узлов, начиная с момента, когда x стал дочерним узлом иного узла.

Дочерние узлы x объединены при помощи указателей $left$ и $right$ в один циклический дважды связанный список дочерних узлов x .

Корни всех деревьев в H связаны при помощи указателей $left$ и $right$ в циклический дважды связанный список корней.

Обращение к H выполняется посредством указателя $min[H]$ на корень дерева с минимальным ключом. Этот узел называется *минимальным узлом* H .

Текущее количество узлов в H хранится в $n[H]$.

На рис. 1 приведен пример фибоначчиевой пирамиды.

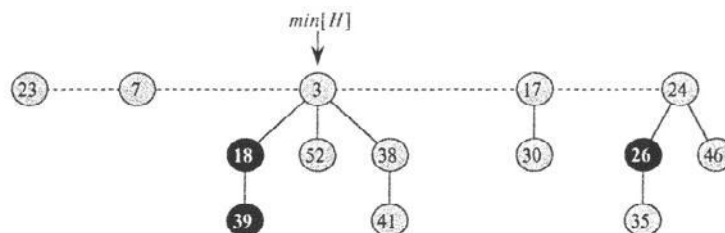


Рисунок 1 – Фибоначчиева пирамида, состоящая из 5 деревьев и 14 узлов

ОПЕРАЦИИ В ФИБОНАЧЧИЕВЫХ ПИРАМИДАХ

Создание пустой фибоначчиевой пирамиды

Процедура создания пустой фибоначчиевой пирамиды H выделяет память и возвращает объект фибоначчиевой пирамиды H , причем $n[H] = 0$, $min[H] = NIL$.

Вставка узла в фибоначчиеву пирамиду

Процедура вставки узла x в фибоначчиевую пирамиду H делает x деревом из одного узла в составе H .

Объединение двух фибоначчиевых пирамид

Процедура объединения двух фибоначчиевых пирамид H_1 и H_2 объединяет H_1 и H_2 , соединяя списки корней H_1 и H_2 , находя затем новый минимальный узел.

Поиск минимального узла

Процедура поиска минимального узла в фибоначчиевой пирамиде H возвращает указатель $min[H]$.

Извлечение минимального узла

Процедура извлечения минимального узла перемещает в список корней все дочерние узлы минимального узла и удаляет последний из списка корней. Затем выполняется уплотнение списка корней путем связывания корней одинаковой степени, пока в списке останется не больше одного корня каждой степени.

Уменьшение ключа

Выполнение процедуры уменьшения ключа узла x до k может нарушить свойство неубывающих пирамид. Его восстановление выполняется при помощи операций вырезания и каскадного вырезания узла, выполняющих перенос x в список корней H , а также перенос отмеченной вершины в список корней H соответственно.

Удаление узла

Процедура удаления узла x из фибоначчиевой пирамиды H делает x минимальным узлом H , присваивая его ключу значение $-$, после чего x удаляется из H при помощи процедуры извлечения минимального узла [2].

ТРУДОЕМКОСТЬ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ

Трудоёмкость выполнения операций в d -пирамиде, биномиальной, а также фибоначчиевой пирамиде приведена в таблице 1 [2].

Таблица 1 - Сравнительный анализ выполнения операций в различных структурах данных

Процедура	d -пирамида	Биномиальная пирамида	Фибоначчиева пирамида
Вставка узла	$O(\log_d n)$	$O(\log_2 n)$	$O(1)$
Поиск минимального узла	$O(1)$	$O(\log_2 n)$	$O(1)$
Извлечение минимального узла	$O(d \log_d n)$	$O(\log_2 n)$	$O(\log_2 n)$
Удаление узла	$O(d \log_d n)$	$O(\log_2 n)$	$O(\log_2 n)$
Уменьшение ключа	$O(\log_d n)$	$O(\log_2 n)$	$O(1)$
Объединение двух пирамид	-	$O(\log_2 n)$	$O(1)$
Создание пустой пирамиды	$O(1)$	$O(1)$	$O(1)$

Как следует из таблицы 1, в структуре данных «Фибоначчиева пирамида» операции и процедуры, в которых не требуется удаление, то есть операции вставки, поиска элемента с минимальным ключом и уменьшения ключа, а также создания пустой пирамиды, имеют время работы, равное $O(1)$. Кроме того, «Фибоначчиева пирамида» позволяет также за время $O(1)$ выполнять дополнительную операцию слияния двух пирамид.

Асимптотические границы времени работы операций извлечения минимального узла и удаления узла в фибоначчиевой пирамиде равны $O(\log_2 n)$.

РЕЗУЛЬТАТЫ КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ

Сравнительный анализ трудоёмкости выполнения операций проводился для структур данных «Бинарная пирамида», «Фибоначчиева пирамида». При создании объектов указанных структур данных использовано следующее количество входных данных: 100, 1000, 5000, 25000, 150000 узлов.

Исследование трудоемкости выполнения операций в бинарной и фибоначчевой пирамидах выполнено по следующим параметрам:

- а) количество операций присвоений;
- б) количество операций сравнений;
- в) время выполнения операции.

Результаты компьютерного тестирования приведены на рис. 2-9. Данные рисунки показывают графики изменения трудоемкости выполнения различных операций в зависимости от количества входных данных.

На рис. 2-5 представлены графики изменения количества операций присвоений, а также сравнений, полученных при выполнении операций в рассматриваемых структурах данных. При этом введены следующие обозначения:

- а) ▲, ◆ – количество операций присвоений в бинарной и фибоначчевой пирамидах соответственно;
- б) ✱, + – количество операций сравнения в бинарной и фибоначчевой пирамидах соответственно.

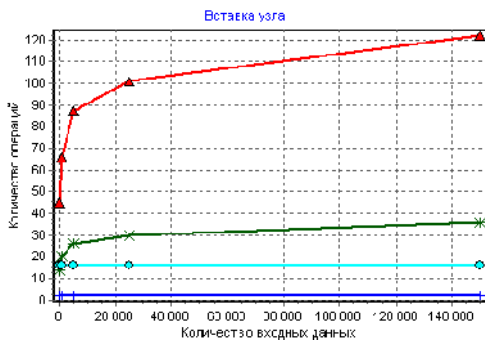


Рисунок 2 – Трудоемкость выполнения операции вставки узла

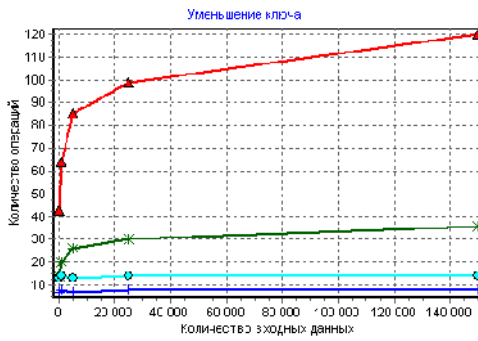


Рисунок 3 – Трудоемкость выполнения операции уменьшения ключа

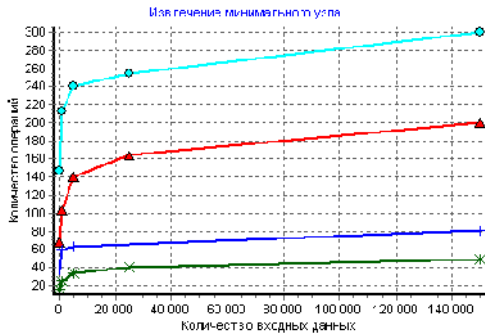


Рисунок 4 – Трудоемкость выполнения операции извлечения минимального узла

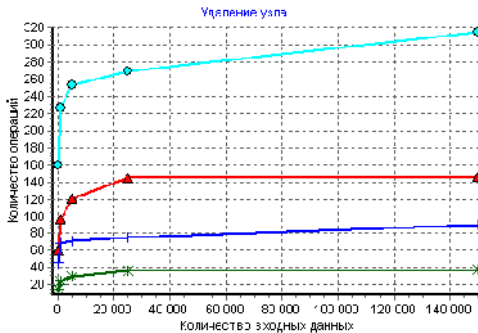


Рисунок 5 – Трудоемкость выполнения операции удаления узла

Процедуры слияния двух фибоначчевых пирамид, а также создания пустой пирамиды и поиска минимального узла в рассматриваемых структурах данных имеют время выполнения $O(1)$, то есть не зависящее от количества входных данных.

На рис. 6-9 приведены графики изменения времени выполнения операций вставки узла, уменьшения ключа, извлечения минимального узла и удаления узла. При этом введены следующие обозначения:

- а) ◆ – время выполнения операции в бинарной пирамиде;
- б) ▲ – время выполнения операции в фибоначчевой пирамиде.

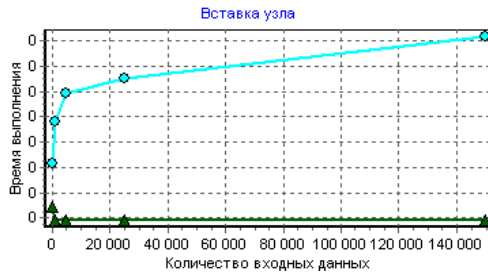


Рисунок 6 – Время выполнения операции вставки узла

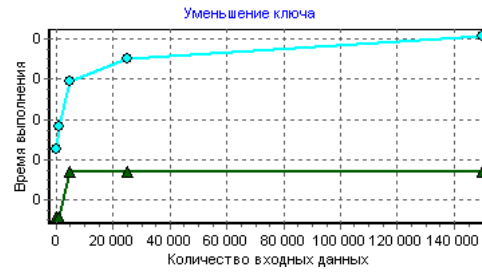


Рисунок 7 – Время выполнения операции уменьшения ключа

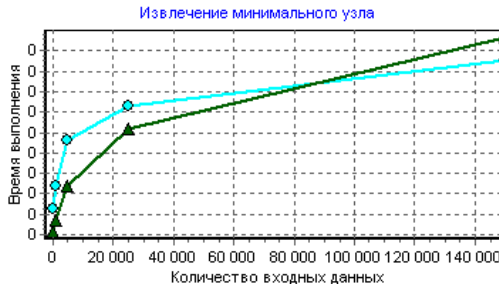


Рисунок 8 – Время выполнения операции извлечения минимального узла

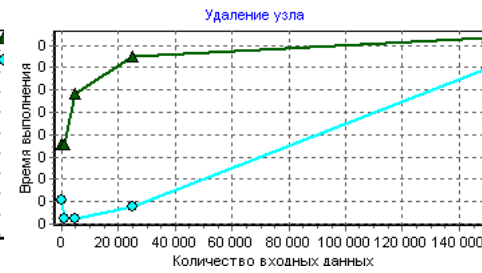


Рисунок 9 – Время выполнения операции удаления узла

Время выполнения операций создания пустой пирамиды, поиска минимального узла в бинарной и фибоначчиевой пирамидах, а также объединения двух фибоначчиевых пирамид не зависит от количества входных данных.

Таким образом, из проведенного компьютерного тестирования следует, что:

а) время выполнения операций создания пустой бинарной и фибоначчиевой пирамид, поиска минимального узла в рассматриваемых структурах данных, а также объединения двух фибоначчиевых пирамид составляет $O(1)$;

б) трудоемкость выполнения операции вставки узла, уменьшения ключа в бинарной пирамиде составляет $O(\log_2 n)$ (рис. 2-3, 6-7 явно показывают логарифмический рост), в фибоначчиевой – $O(1)$;

в) время выполнения операций извлечения минимального узла и удаления узла в фибоначчиевой пирамиде, составившее $O(\log_2 n)$ (см. рисунки 4-5, 8-9), превышает время выполнения данных операций в бинарной пирамиде, асимптотические границы времени выполнения которых составляют $O(2 \log_2 n)$ (см. рисунки 4-5, 8-9).

ВЫВОДЫ

Проведено компьютерное тестирование трудоемкости выполнения операций в бинарной и фибоначчиевой пирамидах. Показано, что трудоемкость выполнения операций в рассматриваемых структурах данных соответствует теоретическим оценкам, приведенным в таблице 1.

Эффективное применение фибоначчиевых пирамид возможно в таких задачах, где:

– количество операций извлечения минимального узла и удаления узла относительно мало по сравнению с количеством других операций;
– время выполнения операции уменьшения ключа $O(1)$ представляет собой существенный выигрыш по сравнению со временем $O(\log_d n)$ в d -пирамидах.

SUMMARY

COMPUTER TESTING OF ALGORITHMS BASED ON FIBONACCI NUMBERS

*Shapovalov S. P., Stadnik A. A.,
Sumy State University, Sumy*

A data structure "Fibonacci heap" was implemented. A research of complexity of operations in this data structure was compared with binary heaps.

Key words: a data structure "Fibonacci heap", complexity of operations.

СПИСОК ЛИТЕРАТУРЫ

1. Стахов А. П. Три «ключевые» проблемы математики на этапе ее зарождения и новые направления в развитии математики, теоретической физики и информатики – <http://www.trinitas.ru/rus/doc/0232/004a/02321064.htm>
2. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л. Алгоритмы: построение и анализ. - 2-е издание; пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 1296 с.
3. Ворожцов А. В. Алгоритмы: построение, анализ и реализация на языке программирования Си: лекции / А. В. Ворожцов, Н. А. Винокуров. – М.: МФТИ, 2007. – 452 с.
4. Кормен Т. Х. Алгоритмы: построение и анализ, 2-е издание: пер. с англ. / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест. – М.: Издательский дом «Вильямс», 2005. – 1296 с.

Поступила в редакцию 12 июля 2011 г.