

Фибоначчиева куча

Материал из Википедии — свободной энциклопедии

Фибоначчиева куча (англ. *Fibonacci heap*) — структура данных, представляющая собой набор деревьев, упорядоченных в соответствии со свойством неубывающей пирамиды. Фибоначчиевы кучи были введены Майклом Фредманом и Робертом Тарьяном в 1984 году.

Структура является реализацией абстрактного типа данных «Очередь с приоритетом», и замечательна тем, что операции, в которых не требуется удаление, имеют амортизированное время работы, равное $O(1)$ (для двоичной кучи и биномиальной кучи амортизационное время работы равно $O(\log n)$). Кроме стандартных операций INSERT, MIN, DECREASE-KEY, фибоначчиева куча позволяет за время $O(1)$ выполнять операцию UNION слияния двух куч.

Содержание

Структура

Операции

- Создание новой фибоначчиевой кучи
- Вставка узла
- Поиск минимального узла
- Объединение двух фибоначчиевых куч
- Извлечение минимального узла
- Уменьшение ключа
- Удаление узла

См. также

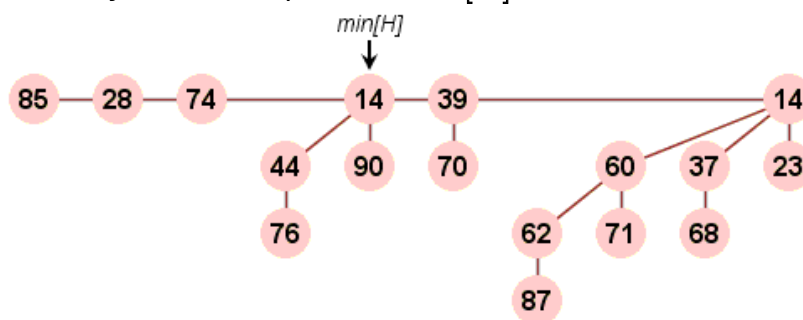
Ссылки

Литература

Структура

- Фибоначчиева куча ***H*** представляет собой набор деревьев.
- Каждое дерево в ***H*** подчиняется свойству кучи (англ. *min-heap property*): ключ каждого узла не меньше ключа его родительского узла.
- Каждый узел ***x*** в ***H*** содержит следующие указатели и поля:
 - key***[***x***] — поле, в котором хранится ключ;
 - p***[***x***] — указатель на родительский узел;
 - child***[***x***] — указатель на один из дочерних узлов;
 - left***[***x***] — указатель на левый сестринский узел;
 - right***[***x***] — указатель на правый сестринский узел;
 - degree***[***x***] — поле, в котором хранится количество дочерних узлов;

- $mark[x]$ — логическое значение, которое указывает, были ли потери узлом x дочерних узлов, начиная с момента, когда x стал дочерним узлом какого-то другого узла.
- Дочерние узлы x объединены при помощи указателей $left$ и $right$ в один циклический двусвязный список дочерних узлов (англ. *child list*) x .
- Корни всех деревьев в H связаны при помощи указателей $left$ и $right$ в циклический двусвязный список корней (англ. *root list*).
- Для всей Фибоначчиевой кучи также хранится указатель на узел с минимальным ключом $min[H]$, являющийся корнем одного из деревьев. Этот узел называется минимальным узлом (англ. *minimum node*) H .
- Текущее количество узлов в H хранится в $n[H]$.



Операции

Создание новой фибоначчиевой кучи

Процедура `Make_Fib_Heap` возвращает объект фибоначчиевой кучи H , $n[H] = 0$ и $min[H] = \text{NULL}$. Деревьев в H нет.

Амортизированная стоимость процедуры равна её фактической стоимости $O(1)$.

Вставка узла

```

Fib_Heap_Insert( $H, x$ )
1  $degree[x] \leftarrow 0$ 
2  $p[x] \leftarrow \text{NULL}$ 
3  $child[x] \leftarrow \text{NULL}$ 
4  $left[x] \leftarrow x$ 
5  $right[x] \leftarrow x$ 
6  $mark[x] \leftarrow \text{FALSE}$ 
7 Присоединение списка корней, содержащего  $x$ , к списку корней  $H$ 
8 if  $min[H] = \text{NULL}$  или  $key[x] < key[min[H]]$ 
9   then  $min[H] \leftarrow x$ 
10  $n[H] \leftarrow n[H] + 1$ 

```

Амортизированная стоимость процедуры равна её фактической стоимости $O(1)$.

Поиск минимального узла

Процедура `Fib_Heap_Minimum` возвращает указатель $min[H]$.

Амортизированная стоимость процедуры равна её фактической стоимости $O(1)$.

Объединение двух фибоначчиевых куч

```

Fib_Heap_Union( $H_1, H_2$ )
1  $H \leftarrow \text{Make\_Fib\_Heap}()$ 
2  $\text{min}[H] \leftarrow \text{min}[H_1]$ 
3 Добавление списка корней  $H_2$  к списку корней  $H$ 
4 if ( $\text{min}[H_1] = \text{NULL}$ ) или ( $\text{min}[H_2] \neq \text{NULL}$  и  $\text{key}[\text{min}[H_2]] < \text{key}[\text{min}[H_1]]$ )
5     then  $\text{min}[H] \leftarrow \text{min}[H_2]$ 
6  $n[H] \leftarrow n[H_1] + n[H_2]$ 
7 Освобождение объектов  $H_1$  и  $H_2$ 
8 return  $H$ 

```

Амортизированная стоимость процедуры равна её фактической стоимости $O(1)$.

Извлечение минимального узла

```

Fib_Heap_Extract_Min( $H$ )
1  $z \leftarrow \text{min}[H]$ 
2 if  $z \neq \text{NULL}$ 
3     then for для каждого дочернего по отношению к  $z$  узла  $x$ 
4         do Добавить  $x$  в список корней  $H$ 
5          $p[x] \leftarrow \text{NULL}$ 
6         Удалить  $z$  из списка корней  $H$ 
7         if  $z = \text{right}[z]$ 
8             then  $\text{min}[H] \leftarrow \text{NULL}$ 
9             else  $\text{min}[H] \leftarrow \text{right}[z]$ 
10        Consolidate( $H$ )
11     $n[H] \leftarrow n[H] - 1$ 
12 return  $z$ 

```

На одном из этапов операции извлечения минимального узла выполняется уплотнение (англ. *consolidating*) списка корней H . Для этого используется вспомогательная процедура Consolidate. Эта процедура использует вспомогательный массив $A[0..D[n[H]]]$. Если $A[i] = y$, то y в настоящий момент является корнем со степенью $\text{degree}[y] = i$.

```

Consolidate( $H$ )
1 for  $i \leftarrow 0$  to  $D(n[H])$ 
2     do  $A[i] \leftarrow \text{NULL}$ 
3 for для каждого узла  $w$  в списке корней  $H$ 
4     do  $x \leftarrow w$ 
5      $d \leftarrow \text{degree}[x]$ 
6     while  $A[d] \neq \text{NULL}$ 
7         do  $y \leftarrow A[d]$  //Узел с той же степенью, что и у  $x$ 
8         if  $\text{key}[x] > \text{key}[y]$ 
9             then обменять  $x \leftrightarrow y$ 
10        Fib_Heap_Link( $H, y, x$ )
11         $A[d] \leftarrow \text{NULL}$ 
12         $d \leftarrow d + 1$ 
13     $A[d] \leftarrow x$ 
14  $\text{min}[H] \leftarrow \text{NULL}$ 
15 for  $i \leftarrow 0$  to  $D(n[H])$ 
16     do if  $A[i] \neq \text{NULL}$ 
17         then Добавить  $A[i]$  в список корней  $H$ 
18         if  $\text{min}[H] = \text{NULL}$  или  $\text{key}[A[i]] < \text{key}[\text{min}[H]]$ 
19             then  $\text{min}[H] \leftarrow A[i]$ 

```

```

Fib_Heap_Link( $H, y, x$ )
1 Удалить  $y$  из списка корней  $H$ 
2 Сделать  $y$  дочерним узлом  $x$ , увеличить  $\text{degree}[x]$ 
3  $\text{mark}[y] \leftarrow \text{FALSE}$ 

```

Амортизированная стоимость извлечения минимального узла равна $O(\log n)$.

Уменьшение ключа

```

Fib_Heap_Decrease_Key( $H, x, k$ )
1 if  $k > \text{key}[x]$ 

```

```

2  then error «Новый ключ больше текущего»
3   $key[x] \leftarrow k$ 
4   $y \leftarrow p[x]$ 
5  if  $y \neq \text{NULL}$  и  $key[x] < key[y]$ 
6  then Cut( $H, x, y$ )
7      Cascading_Cut( $H, y$ )
8  if  $key[x] < key[\min[H]]$ 
9  then  $\min[H] \leftarrow x$ 

```

```

Cut( $H, x, y$ )
1  Удаление  $x$  из списка дочерних узлов  $y$ , уменьшение  $degree[y]$ 
2  Добавление  $x$  в список корней  $H$ 
3   $p[x] \leftarrow \text{NULL}$ 
4   $mark[x] \leftarrow \text{FALSE}$ 

```

```

Cascading_Cut( $H, y$ )
1   $z \leftarrow p[y]$ 
2  if  $z \neq \text{NULL}$ 
3  then if  $mark[y] = \text{FALSE}$ 
4      then  $mark[y] \leftarrow \text{TRUE}$ 
5      else Cut( $H, y, z$ )
6      Cascading_Cut( $H, z$ )

```

Амортизированная стоимость уменьшения ключа не превышает $O(1)$.

Удаление узла

```

Fib_Heap_Delete( $H, x$ )
1  Fib_Heap_Decrease_Key( $H, x, -\infty$ )
2  Fib_Heap_Extract_Min( $H$ )

```

Амортизированное время работы процедуры равно $O(\log n)$.

См. также

- Очередь с приоритетом (программирование)
- Двоичная куча
- Биномиальная куча

Ссылки

- Реализация структуры на C (https://web.archive.org/web/20070701192433/http://resnet.uoregon.edu/~gurney_j/jmpc/fib.html) (англ.)
- Владимир Алексеев, Владимир Таланов, Лекция 7: Биномиальные и фибоначчиевы кучи (<http://www.intuit.ru/studies/courses/100/100/lecture/1541?page=2>) // "Структуры данных и модели вычислений", 26.09.2006, intuit.ru

Литература

- Томас Х. Кормен и др. Алгоритмы: построение и анализ (https://archive.org/details/introductiontoal0000unse_h3k5/page/1296). — 2-е изд. — М.: Издательский дом «Вильямс», 2007. — С. 1296. — ISBN 5-8459-0857-4.
- Mehlhorn, Kurt, Sanders, Peter. 6.2.2 Fibonacci Heaps // Algorithms and Data Structures: The Basic Toolbox. — Springer, 2008. — 300 с. — ISBN 978-3-540-77978-0.

Источник — https://ru.wikipedia.org/w/index.php?title=Фибоначчиева_куча&oldid=137236941

Эта страница в последний раз была отредактирована 13 апреля 2024 в 18:55.

Текст доступен по лицензии Creative Commons «С указанием авторства — С сохранением условий» (CC BY-SA); в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации «Фонд Викимедиа» (Wikimedia Foundation, Inc.)