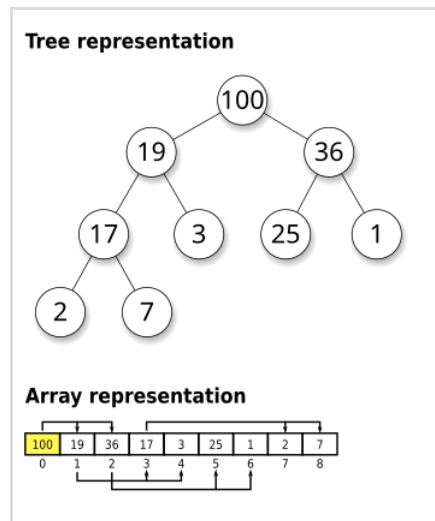


Куча (структура данных)

Материал из Википедии — свободной энциклопедии

Ку́ча (англ. *heap*) в программировании — специализированная структура данных типа дерева, которая удовлетворяет *свойству кучи*: если ***B*** является узлом-потомком узла ***A***, то ***k**(**A**) ≥ **k**(**B**)*, где ***k**(**X**)* — ключ (идентификатор) узла. Из этого следует, что элемент с наибольшим значением ключа всегда является корневым узлом кучи, поэтому иногда такие кучи называют *тах-кучами* (в качестве альтернативы, если сравнение перевернуть, то наименьший элемент будет всегда корневым узлом, такие кучи называют *min-кучами*). Не существует никаких ограничений относительно того, сколько узлов-потомков имеет каждый узел кучи, хотя на практике их число обычно не более двух. Куча является максимально эффективной реализацией абстрактного типа данных, который называется очередью с приоритетом. Кучи имеют решающее значение в некоторых эффективных алгоритмах на графах, таких, как алгоритм Дейкстры на d-кучах и сортировка методом пирамиды.



Пример полной двоичной кучи

В ранних реализациях Лиспа обеспечивалось динамическое распределение памяти с использованием кучи как структуры данных, впоследствии всякую динамически распределяемую память стали называть «кучей» (хотя она не обязательно использует соответствующую структуру) [1].

Кучи обычно реализуются в виде массивов, что исключает наличие указателей между её элементами.

Над кучами обычно проводятся следующие операции:

- *найти максимум* или *найти минимум*: найти максимальный элемент в тах-куче или минимальный элемент в min-куче, соответственно
- *удалить максимум* или *удалить минимум*: удалить корневой узел в тах- или min-куче, соответственно
- *увеличить ключ* или *уменьшить ключ*: обновить ключ в тах- или min-куче, соответственно
- *добавить*: добавление нового ключа в кучу.
- *слияние*: соединение двух куч с целью создания новой кучи, содержащей все элементы обеих исходных.

Содержание

Варианты

Применение

Реализации

Примечания

Варианты

В зависимости от ограничений на структуры используются различные варианты куч, некоторые из них:

- 2-3-куча
- Двуродительская куча
- Двоичная куча
- Биномиальная куча
- Очередь Бродала (очередь с параллельным приоритетом)^[2]
- Куча с D потомками
- Фибоначчиева куча
- Куча с приоритетом самого левого
- Спаренная куча
- Асимметричная куча
- Мягкая куча
- Тернарная куча
- Декартово дерево

Различные варианты демонстрируют различную временную сложность вычислений для различных операций^[1] (имена операций в нотации для min-кучи):

Операция	Двоичная	Биномиальная	Фибоначчиева	Спаренная ^[3]	Бродала
найти минимум	$\Theta(1)$	$\Theta(\log n)$ или $\Theta(1)$	$\Theta(1)$ ^[1]	$\Theta(1)$	$\Theta(1)$
удалить минимум	$\Theta(\log n)$	$\Theta(\log n)$	$O(\log n)^*$	$O(\log n)^*$	$O(\log n)$
добавить	$\Theta(\log n)$	$O(\log n)$	$\Theta(1)$	$O(1)^*$	$\Theta(1)$
уменьшить ключ	$\Theta(\log n)$	$\Theta(\log n)$	$\Theta(1)^*$	$O(\log n)^*$	$\Theta(1)$
слияние	$\Theta(n)$	$O(\log n)^{**}$	$\Theta(1)$	$O(1)^*$	$\Theta(1)$

(*) Амортизационное время — оценка сделана методом амортизационного анализа (по наихудшему времени), в остальных случаях оценка является регулярным худшим случаем.

(**) n — размер наибольшей кучи

Здесь $O(F)$ даёт асимптотическую верхнюю границу, а $\Theta(F)$ является асимптотически точной оценкой (в соответствии с нотацией « O » большое и « o » малое).

Применение

Структуры данных типа кучи имеют множество применений.

Пирамидальная сортировка: один из лучших применяемых методов сортировки, не имеющий квадратичных наихудших сценариев.

Алгоритмы поиска: при использовании кучи поиск минимума, максимума, того и другого, медианы или k -го наибольшего элемента может быть сделан за линейное время (часто даже за константное время).^[4]

Алгоритмы на графах: применение кучи в качестве структуры данных для внутреннего обхода даёт сокращение времени выполнения на полиномиальный порядок. Примерами таких проблем являются алгоритм построения минимального остовного дерева Прима и проблема кратчайшего пути Дейкстры.

Полная и почти полная бинарная куча может быть представлена очень эффективным способом с помощью индексного массива. Первый (или последний) элемент будет содержать корень. Следующие два элемента массива содержат узлы-потомки корня. Следующие четыре элемента содержат четверых потомков от двух узлов — потомков корня, и так далее. Таким образом, потомки узла уровня n будут расположены на позициях $2n$ и $2n+1$ для массива, индексируемого с единицы, или на позициях $2n+1$ и $2n+2$ для массива, индексируемого с нуля. Это позволяет перемещаться вверх или вниз по дереву, выполняя простые вычисления индекса массива. Балансировка кучи делается перестановкой элементов, которые нарушают порядок. Поскольку мы можем построить кучу с помощью массива без дополнительной памяти (для узлов, например), то можно использовать пирамидальную сортировку для сортировки массива прямо на месте.

Реализации

Стандартная библиотека шаблонов языка C++ предоставляет шаблоны функций для управления кучей `make_heap`, `push_heap` и `pop_heap` (обычно реализуются бинарные кучи), которые оперируют с итераторами произвольного доступа. Методы используют итераторы как ссылки на массивы и выполняют преобразование массив-куча.

Набор шаблонов Java платформы Java 2 (начиная с версии 1.5) предоставляет реализацию бинарной кучи в классе `java.util.PriorityQueue<E>`.

Python имеет модуль `heapq`, который реализует очереди с приоритетами с помощью бинарной кучи^[5].

Начиная с версии 5.3 PHP в стандартной библиотеке имеются методы `maxheap` (`SplMaxHeap`) и `minheap` (`SplMinHeap`).

В Perl имеются реализации бинарной, биномиальной и фибоначчиевой кучи во всеобъемлющей сети архивов^[6].

Примечания

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest (1990): Introduction to algorithms. MIT Press / McGraw-Hill.
2. *A Parallel Priority Queue with Constant Time Operations* (<http://www.ceid.upatras.gr/faculty/zaro/pub/jou/J9-JPDC-pq.pdf>) (PDF), Архивировано (<https://web.archive.org/web/20110726093807/http://www.ceid.upatras.gr/faculty/zaro/pub/jou/J9-JPDC-pq.pdf>) (PDF) 26 июля 2011, Дата обращения: 31 мая 2011 {{citation}}: Игнорируется текст: "web" (справка) Источник (<https://web.archive.org/web/20110726093807/http://www.ceid.upatras.gr/faculty/zaro/pub/jou/J9-JPDC-pq.pdf>) . Дата обращения: 31 мая 2011. Архивировано 26 июля 2011 года.
3. Iacono, John (2000), "Improved upper bounds for pairing heaps", *Proc. 7th Scandinavian Workshop on Algorithm Theory*, Lecture Notes in Computer Science, vol. 1851, Springer-Verlag, pp. 63—77, doi:[10.1007/3-540-44985-X_5](https://doi.org/10.1007/3-540-44985-X_5) (https://doi.org/10.1007/3-540-44985-X_5)
4. Frederickson, Greg N. (1993), "An Optimal Algorithm for Selection in a Min-Heap", *Information and Computation* (https://web.archive.org/web/20121203045606/http://ftp.cs.purdue.edu/research/technical_reports/1991/TR%2091-027.pdf) (PDF), vol. 104,

Academic Press, pp. 197—214, doi:10.1006/inco.1993.1030 (<https://doi.org/10.1006%2Finc o.1993.1030>), Архивировано из оригинала (http://ftp.cs.purdue.edu/research/technical_reports/1991/TR%2091-027.pdf) (PDF) 3 декабря 2012, Дата обращения: 31 мая 2011
Источник (https://web.archive.org/web/20121203045606/http://ftp.cs.purdue.edu/research/technical_reports/1991/TR%2091-027.pdf) . Дата обращения: 31 мая 2011.
Архивировано 3 декабря 2012 года.

5. Python heapq (<https://docs.python.org/library/heapq.html>). Дата обращения: 31 мая 2011. Архивировано (<https://web.archive.org/web/20121018042235/http://docs.python.org/library/heapq.html>) 18 октября 2012 года.
6. Perl Heap (<https://search.cpan.org/perl/doc?Heap>)

Источник — [https://ru.wikipedia.org/w/index.php?title=Куча_\(структура_данных\)&oldid=138133670](https://ru.wikipedia.org/w/index.php?title=Куча_(структура_данных)&oldid=138133670)

Эта страница в последний раз была отредактирована 31 мая 2024 в 22:39.

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.)