ВикипедиЯ

Очередь с приоритетом (программирование)

Материал из Википедии — свободной энциклопедии

Очередь с приоритетом (англ. *priority queue*) — абстрактный тип данных в программировании, поддерживающий две обязательные операции — добавить элемент и извлечь максимум[1](минимум). Предполагается, что для каждого элемента можно вычислить его *приоритет* — действительное число или в общем случае элемент линейно упорядоченного множества[2].

Содержание

Описание

Основные методы, реализуемые очередью с приоритетом, следующие[2][3][1]:

- `insert(ключ, значение)` — добавляет пару `(ключ, значение)` в хранилище;
- `extract_minimum()` — возвращает пару `(ключ, значение)` с минимальным значением ключа, удаляя её из хранилища.

При этом меньшее значение ключа соответствует более высокому приоритету.

В некоторых случаях более естественен рост ключа вместе с приоритетом. Тогда второй метод можно назвать `extract_maximum()`[1].

Есть ряд реализаций в которых обе основные операции выполняются в худшем случае за время, ограниченное $O(\log n)$ (см. «O» большое и «o» малое), где n — количество хранимых пар.

Примеры

В качестве примера очереди с приоритетом можно рассмотреть список задач работника. Когда он заканчивает одну задачу, он переходит к очередной — самой приоритетной (ключ будет величиной, обратной приоритету) — то есть выполняет операцию извлечения максимума. Начальник добавляет задачи в список, указывая их приоритет, то есть выполняет операцию добавления элемента.

Расширения очереди с приоритетом

На практике интерфейс очереди с приоритетом нередко расширяют другими операциями[4][3]:

- вернуть минимальный элемент без удаления из очереди
- изменить приоритет произвольного элемента
- удалить произвольный элемент
- слить две очереди в одну

В ***индексированных очередях с приоритетом*** (адресуемых[5]) возможно обращение к элементам по индексу. Такие очереди могут быть использованы, например, для слияния нескольких отсортированных последовательностей (multiway merge)[1].

Могут также рассматриваться ***очереди с приоритетом с двусторонним доступом*** (англ. *double-ended priority queue, DEPQ*), в которых есть операции доступа как к минимальному, так и к максимальному элементу[6].

Реализации

Очередь с приоритетами может быть реализована на основе различных структур данных.

Простейшие (и не очень эффективные) реализации могут использовать неупорядоченный или упорядоченный массив, связный список, подходящие для небольших очередей. При этом вычисления могут быть как «ленивыми» (тяжесть вычислений переносится на извлечение элемента), так и ранними (eager), когда вставка элемента сложнее его извлечения. То есть, одна из операций может быть произведена за время $O(1)$, а другая — в худшем случае за $O(N)$, где N — длина очереди[1].

Более эффективными являются реализации на основе кучи, где обе операции можно производить в худшем случае за время $O(\log N)$[1]. К ним относятся двоичная куча, биномиальная куча, фибоначчиева куча, pairing heap[6].

Абстрактный тип данных (АТД) для очереди с приоритетом получается из АТД кучи переименованием соответствующих функций. Минимальное (максимальное) значение находится всегда на вершине кучи[7].

Пример на Python

Стандартная библиотека Python содержит модуль `heapq`[8], реализующий очередь с приоритетом[9]:

```
# импорт двух функций очереди под именами, принятыми в данной статье
from heapq import heappush as insert, heappop as extract_maximum
pq = []  # инициализация списка
insert(pq, (4, 0, "p"))   # вставка в очередь элемента "p" с индексом 0 и приоритетом 4
insert(pq, (2, 1, "e"))
insert(pq, (3, 2, "a"))
insert(pq, (1, 3, "h"))
# вывод четырёх элементов в порядке возрастания приоритетов
print(extract_maximum(pq)[-1] + extract_maximum(pq)[-1] + extract_maximum(pq)[-1] + extract_maximum(pq)[-1])
```

Этот пример выведет слово «heap».

Примечания

1. Sedgewick, Wayne, 2011.
2. Ахо, Хопкрофт, Ульман, 2000.
3. Кормен и др., 2005.
4. *Robert Sedgewick.* Algorithms in C++, Parts 1–4: Fundamentals, Data Structure, Sorting, Searching. — Third Edition. — Addison-Wesley Professional. — 752 p. — ISBN 978-0-7686-8533-6.
5. Mehlhorn, Sanders, 2008.
6. Sahni, Mehta, 2018.

7. Rabhi, Lapalme, 1999.
8. 8.4. heapq — Heap queue algorithm (https://docs.python.org/2/library/heapq.html)
9. *David Beazley, Brian K. Jones.* 1.5. Implementing a Priority Queue // Python Cookbook. — 3rd Edition. — O'Reilly Media, Inc., 2013. — 706 p. — ISBN 978-1-4493-4037-7.

Литература

- *Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К.* Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И. В. Красикова. — 2-е изд.. — М.: Вильямс, 2005. — 1296 с. — ISBN 5-8459-0857-4.
- *Ахо А. В, Хопкрофт Дж. Э., Ульман Дж. Д.* Структуры данных и алгоритмы = Data Structures and Algorithms. — Вильямс, 2000. — 384 с. — ISBN 9785845901224., *4.10. Очереди с приоритетами*
- *Robert Sedgewick; Kevin Wayne.* 2.4 Priority Queues // Algorithms. — Fourth Edition. — Addison-Wesley Professional, 2011. — 992 с. — ISBN 978-0-13-276257-1.
- *Gerth Stølting Brodal, Chris Okasaki.* Optimal Purely Functional Priority Queues (ftp://ftp.daimi.au.dk/pub/BRICS/Reports/RS/96/37/BRICS-RS-96-37.pdf) // BRICS Report Series. — Department of Computer Science University of Aarhus, 1996. — № RS-96-37. — ISSN 0909-0878 (https://www.worldcat.org/search?fq=x0:jrnl&q=n2:0909-0878).
- *Fethi Rabhi and Lapalme, G.* Algorithms: A Functional Programming Approach. — Addison-Wesley, 1999. — P. 92—93, 106-107. — 235 p. — ISBN 9780201596045.
- *Sartaj Sahni and Dinesh P. Mehta.* Part II: Priority Queues // Handbook of Data Structures and Applications. — 2nd ed. — Chapman and Hall/CRC, 2018. — 1100 p. — ISBN 9781498701853.
- *Mehlhorn, Kurt, Sanders, Peter.* 6. Priority Queues // Algorithms and Data Structures: The Basic Toolbox. — Springer, 2008. — 300 с. — ISBN 978-3-540-77978-0.

Ссылки

- C++ STL `priority_queue`:
  - страница помощи std::priority_queue на MSDN (http://msdn.microsoft.com/en-us/library/6w541ec5.aspx)
  - страница помощи std::priority_queue на SGI (http://www.sgi.com/tech/stl/priority_queue.html)
- Очереди с приоритетом для Ruby:
  - Brian Amberg's priority-queue (http://ruby.brian-amberg.de/priority-queue)
  - PriorityQueue в Ruby Application Archive (http://raa.ruby-lang.org/project/priorityqueue)
- Верифицированная с помощью Coq реализация очереди с приоритетом для Haskell:
  - Implementation and verification of priority queues (https://code.google.com/p/priority-queues/)
