

第 6 章 优先队列(堆)

虽然发送到打印机的作业一般被放到队列中,但这未必总是最好的做法。例如,可能有一项作业特别重要,因此希望只要打印机一有空闲就来处理这项作业。反之,若在打印机有空时正好有多个单页的作业及一项 100 页的作业等待打印,则更合理的做法也许是最后处理长的作业,尽管它不是最后提交上来的(不幸的是,大多数的系统并不这么做,有时可能特别令人懊恼)。

类似地,在多用户环境中,操作系统调度程序必须决定在若干进程中运行哪个进程。一般一个进程只被允许运行一个固定的时间片。一种算法是使用一个队列。开始时作业被放到队列的末尾。调度程序将反复提取队列中的第一个作业并运行它,直到运行完毕,或者该作业的时间片用完,并在作业未运行完毕时把它放到队列的末尾。这种策略一般并不太合适,因为一些很短的作业由于一味等待运行而要花费很长的时间去处理。一般说来,短的作业要尽可能快地结束,这一点很重要,因此在已经运行的作业当中这些短作业应该拥有优先权。此外,有些作业虽不短小但很重要,也应该拥有优先权。

这种特殊的应用似乎需要一类特殊的队列,我们称之为**优先队列**(priority queue)。在本章中,我们将讨论:

- 优先队列 ADT 的有效实现。
- 优先队列的使用。
- 优先队列的高级实现。

我们将看到的这类数据结构属于计算机科学中最精致的一种。

6.1 模型

优先队列是允许至少下列两种操作的数据结构:insert(插入),它的作用是显而易见的;以及deleteMin(删除最小者),它的工作是找出、返回并删除优先队列中最小的元素。insert 操作等价于 enqueue(入队),而 deleteMin 则是队列运算 dequeue(出队)在优先队列中的等价操作。

如同大多数数据结构那样,有时可能要添加一些其他的操作,但这些添加的操作属于扩展的操作,而不是图 6-1 所描述的基本模型的一部分。

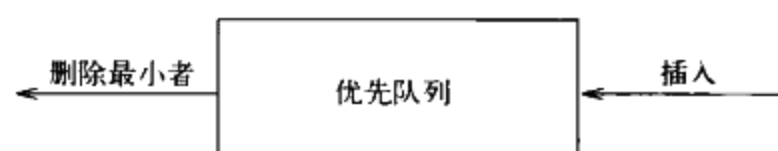


图 6-1 优先队列的基本模型

除了操作系统外,优先队列还有许多的应用。在第 7 章,我们将看到优先队列如何用于外部排序。在**贪婪算法**(greedy algorithm)的实现方面优先队列也是很重要的,该算法通过反复求出最小元来进行操作;在第 9 章和第 10 章我们将看到一些特殊的例子。本章将介绍优先队列在离散事件模拟中的一个应用。