堆 (数据结构)

堆（英语：Heap）是计算机科学中一类特殊的数据结构的统称。堆通常是一个可以被看做一棵树的数组对象。在队列中，调度程序反复提取队列中第一个作业并运行，因为实际情况中某些时间较短的任务将等待很长时间才能结束，或者某些不短小，但具有重要性的作业，同样应当具有优先权。堆即为解决此类问题设计的一种数据结构。

逻辑定义

n个元素序列{k_1,k_2...k_i...k_n},当且仅当满足下列关系时称之为堆：
(k_i <= k_{2i},k_i <= k_{2i+1})或者(k_i >= k_{2i},k_i >= k_{2i+1}), (i = 1,2,3,4...n/2)

性质

堆的实现通过构造**二叉堆**（binary heap），实为二叉树的一种；由于其应用的普遍性，当不加限定时，均指该数据结构的这种实现。这种数据结构具有以下性质。

- 任意节点小于（或大于）它的所有后裔，最小元（或最大元）在堆的根上（**堆序性**）。
- 堆总是一棵完全树。即除了最底层，其他层的节点都被元素填满，且最底层尽可能地从左到右填入。

将根节点最大的堆叫做**最大堆**或**大根堆**，根节点最小的堆叫做**最小堆**或**小根堆**。常见的堆有二叉堆、斐波那契堆等。

支持的基本操作

操作	描述	时间复杂度
build	创建一个空堆	O(n)
insert	向堆中插入一个新元素	$O(\log n)$
update	将新元素提升使其匹配堆的性质	
get	获取当前堆顶元素的值	O(1)

操作	描述	时间复杂度
delete	删除堆顶元素	$O(\log n)$
heapify	使删除堆顶元素的堆再次成为堆	

某些堆实现还支持其他的一些操作，如斐波那契堆支持检查一个堆中是否存在某个元素。

例程

为将元素X插入堆中，找到空闲位置，创建一个空穴，若满足**堆序性**（英文：**heap order**），则插入完成；否则将父节点元素装入空穴，删除该父节点元素，完成空穴上移。直至满足堆序性。这种策略叫做**上滤**（percolate up）。[1]

```
void Insert( ElementType X, PriorityQueue H )
{
    int i;

    if( IsFull(H) )
    {
        printf( "Queue is full.\n" );
        return;
    }

    for( i = ++H->Size; H->Element[i/2] > X; i /= 2 )
        H->Elements[i] = H->Elements[i/2];
    H->Elements[i] = X;
}
```

以上是插入到一个二叉堆的过程。

DeleteMin，删除最小元，即二叉树的根或父节点。删除该节点元素后，队列最后一个元素必须移动到堆得某个位置，使得堆仍然满足堆序性质。这种向下替换元素的过程叫作**下滤**。

```
ElementType
DeleteMin( PriorityQueue H )
{
    int i, Child;
    ElementType MinElement, LastElement;
```

```
    if( IsEmpty( H ) )
    {
        printf( "Queue is empty.\n" );
        return H->Elements[0];
    }
    MinElement = H->Elements[1];
    LastElement = H->Elements[H->Size--];

    for( i = 1; i*2 <= H->Size; i = Child )
    {
        /* Find smaller child. */
        Child = i*2;
        if( Child != H->Size && H->Elements[Child+1]
                             <  H->Elements[Child] )
            Child++;

        /* Percolate one level. */
        if( LastElement > H->Elements[Child] )
            H->Elements[i] = H->Elements[Child];
        else
            break;
    }
    H->Elements[i] = LastElement;
    return MinElement;
}
```

应用

堆排序

堆（通常是二叉堆）常用于排序。这种算法称作堆排序。

事件模拟

主要运用堆的排序以选择优先。