

```

void Swap( ElementType *a, ElementType *b )
{
    ElementType t = *a; *a = *b; *b = t;
}

void PercDown( ElementType A[], int p, int N )
{ /* 改编代码4.24的PercDown( MaxHeap H, int p ) */
    /* 将N个元素的数组中以A[p]为根的子堆调整为最大堆 */
    int Parent, Child;
    ElementType X;

    X = A[p]; /* 取出根结点存放的值 */
    for( Parent=p; (Parent*2+1)<N; Parent=Child ) {
        Child = Parent * 2 + 1;
        if( (Child!=N-1) && (A[Child]<A[Child+1]) )
            Child++; /* Child指向左右子结点的较大者 */
        if( X >= A[Child] ) break; /* 找到了合适位置 */
        else /* 下滤X */
            A[Parent] = A[Child];
    }
    A[Parent] = X;
}

void HeapSort( ElementType A[], int N )
{ /* 堆排序 */
    int i;

    for ( i=N/2-1; i>=0; i-- ) /* 建立最大堆 */
        PercDown( A, i, N );

    for ( i=N-1; i>0; i-- ) {
        /* 删除最大堆顶 */
        Swap( &A[0], &A[i] ); /* 见代码7.1 */
        PercDown( A, 0, i );
    }
}

```