

9.3.1选择排序

```
void Selection_Sort(ElementType A[],int N)
{
    for(i=0;i<N;i++)
    {
        MinPosition=ScanForMin(A,i,N-1);
        /* 从A[i]到A[N-1]中找最小元,并将其位置赋值给MinPositon */
        Swap(A[i],A[MinPosition]);
        /* 将未排序部分的最小元换到有序部分的最后位置 */
    }
}
```

9.3.2堆排序

算法1

```
void Heap_Sort(ElementType A[],int N)
{
    BuildHeap(A); /* O(N) */
    for(i=0;i<N;i++)
    {
        TmpA[i]=DeleteMin(A); /* O(NlogN) */
    }
    for(i=0;i<N;i++) /* O(N) */
    {
        A[i]=TmpA[i];
    }
}
```

$$T(N) = O(N\log N)$$

问题在于需要额外 $O(N)$ 空间, 并且复制元素需要时间

算法2

```
void Heap_Sort(ElementType A[],int N)
{
    for(i=N/2;i>=0;i--) /* BuildHeap */
        PercDown(A,i,N);
    for(i=N-1;i>0;i--)
    {
        Swap(&A[0],&A[i]); /* DeleteMax */
        PercDown(A,0,i); /* 该函数用于调整成最大堆 */
    }
}
```

定理：堆排序处理 N 个不同元素的随机排列的平均比较次数是 $2N\log N - O(N\log\log N)$

虽然堆排序给出最佳平均时间复杂度, 但是实际效果不如用Sedgewick增量序列的希尔排序