Algoritmi di “Algoritmi e Strutture Dati”

InseritonSort

Codice

```
InseritonSort(A)
     n = A.lenght
      for j=2 to n
          key = A[j]
          i = j - 1
          while(i>0) and (A[i]>key)
               A[i+1] = A[i]
               i = i -1
          A[i+1] = key
```

Costo

$\sim n^2$

MergeSort

Codice

```
MergeSort(A,p,r)
     if p < r
          q = (p+r)/2
          MergeSort(A,p,q)
          MergeSort(A,q+1,r)
          Merge(A,p,q,r)

Merge(A,p,q,r)
     n1 = q-p+1
     n2 = r-q
     crea L[1...n1+1]
     crea R[1...n2+1]
     for i=1 to n1
          L[i] = A[p+i-1]
     for j=1 to n2
          R[j] = A[q+j]
     L[n1+1] = R[n2+1] = infinity \\elemento sentinella
     i = j = 1
     for k=p to r
          if L[i] <= R[j]
               A[k] = L[i]
               i++
          else
               A[k] = R[j]
               j++
```

Costo

$\sim n \log n$

InsertionSort Ricorsivo

Codice

```
InsertionSort_Rec(A)
     if j>1
          InseritonSort_Rec(A,j-1)
          Insert(A,j)

Insert(A,j)
     if (j>1) and (A[j-1]>A[j])
          A[j-1]<=>A[j]
          Inser(A,j-1)
```

HeapSort

Codice

```
Parent(i)
     return i/2

Letf(i)
     return i*2

Right(i)
     return i*2+1

MaxHeapify(A,i)
     l = Left(i)
     r = Right(i)
     if (l<=A.size) and (A[l]>A[i])
          max = l
     else
          max = i
     if (r<=A.size) and (A[r]>A[max])
          max = r
     if max != i
          A[i]<=>A[max]
          MaxHeapify(A,max)

BuildMaxHeap(A)
     A.size = A.lenght
     for i=(A.size)/2 to 1
          MaxHeapify(A,i)

HeapSort(A)
     BuildMaxHeap(A)
     for i = A.length to 2
          A[1]<=>A[i]
          A.size = A.size-1
          MaxHeapify(A,1)

Max(A)
     return A[1]
```

```
ExtractMax(A)
     max = A[1]
     A[1] = A[size]
     A.size = A.size-1
     MaxHeapify(A,1)

MaxHeapifyUp(A,i)
     if (i>1) and (A[i]>A[parent(i)])
          A[i]<=>A[parent(i)]
          MaxHeapifyUp(A,parent(i))
```

Costo

MaxHeapify = $O(\log n)$
BuildMaxHeap = $O(n \log n)$
HeapSort = $O(n \log n)$
Max = $O(1)$
ExtractMax = $O(\log n)$