Heap

Pseudocodice*

Algoritmo 1 Parent(i)

Parent(i)

1: **return** $i \; div \; 2$

Algoritmo 2 Left(i)

Left(i)

1: **return** $2i$

Algoritmo 3 Right(i)

Right(i)

1: **return** $2i + 1$

*T. H. Cormen, C. E. Leiserson, R. L. Rivest e C. Stein. *Introduction to Algorithms*. MIT Press, Second Edition, 2001.

Algoritmo 4 Heapify(A,i)

Heapify(A,i)

```
1: l ← Left(i)
2: r ← Right(i)
3: if (l ≤ heapsize(A)) and (A[l] > A[i]) then
4:    largest ← l
5: else
6:    largest ← i
7: end if
8: if (r ≤ heapsize(A)) and (A[r] ≥ A[largest]) then
9:    largest ← r
10: end if
11: if largest ≠ i then
12:    Exchange(A, i, largest)
13:    Heapify(A, largest)
14: end if
```

Algoritmo 5 HeapEmpty(A)

HeapEmpty(A)

```
1: if heapsize(A) = 0 then
2:    return TRUE
3: else
4:    return FALSE
5: end if
```

Algoritmo 6 HeapFull(A)

HeapFull(A)

```
1: if heapsize(A) = length(A) then
2:    return TRUE
3: else
4:    return FALSE
5: end if
```

Algoritmo 7 HeapBuild(A)

HeapBuild(A)

```
1: heapsize(A) ← length(A)
2: for i ← length(A) div 2 downto 1 do
3:    Heapify(A, i)
4: end for
```

Algoritmo 8 HeapInsert(A,x)

HeapInsert(A,x)

```
1: if HeapFull(A) then
2:    error OVERFLOW
3: end if
4: heapsize(A) ← heapsize(A) + 1
5: A[heapsize(A)] ← x
6: i ← heapsize(A)
7: while (i > 1) and (A[Parent(i)] < A[i]) do
8:    Exchange(A, i, Parent(i))
9:    i ← Parent(i)
10: end while
```

Algoritmo 9 HeapDelete(A,i)

HeapDelete(A,i)

```
1: if HeapEmpty(A) then
2:    error UNDERFLOW
3: end if
4: old ← A[i]
5: new ← A[heapsize(A)]
6: A[i] ← new
7: heapsize(A) ← heapsize(A) − 1
8: if new < old then
9:    Heapify(A, i)
10: else
11:    while (i > 1) and (A[Parent(i)] < A[i]) do
12:       Exchange(A, i, Parent(i))
13:       i ← Parent(i)
14:    end while
15: end if
```

Algoritmo 10 HeapModifyKey(A,i,k)

HeapModifyKey(A,i,k)

```
1: old ← A[i]
2: new ← k
3: A[i] ← new
4: if new < old then
5:    Heapify(A, i)
6: else
7:    while (i > 1) and (A[Parent(i)] < A[i]) do
8:       Exchange(A, i, Parent(i))
9:       i ← Parent(i)
10:   end while
11: end if
```

Algoritmo 11 ExtractMax(A)

ExtractMax(A)

1: $max \leftarrow HeapMax(A)$
2: $HeapDelete(A, 1)$
3: **return** max

Algoritmo 12 HeapSort(A)

HeapSort(A)

1: $HeapBuild(A)$
2: **for** $i \leftarrow length(A)$ **downto** 2 **do**
3: $\quad Exchange(A, 1, i)$
4: $\quad heapsize(A) \leftarrow heapsize(A) - 1$
5: $\quad Heapify(A, 1)$
6: **end for**