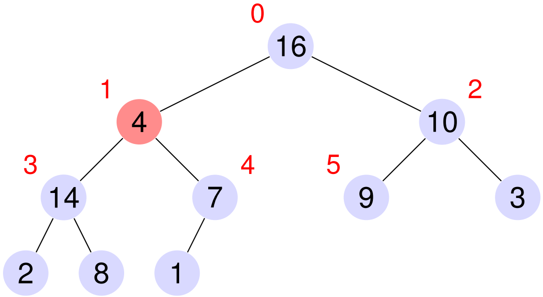
HeapSort

Anna Nenca

```
MAX-HEAPIFY($A$, $heapsize$, $i$)
$l = \mathrm{LEFT}(i)$ // $2i + 1$
$r = \mathrm{RIGHT}(i)$ // $2i + 2$
if $l < heapsize$ and $A[l] > A[i]$
    $largest = l$
else
    $largest = i$
if $r < heapsize$ and $A[r] > A[largest]$
    $largest = r$
if $i \neq largest$
    swap($A[largest]$, $A[i]$)
    MAX-HEAPIFY($A$, $heapsize$, $largest$)
```

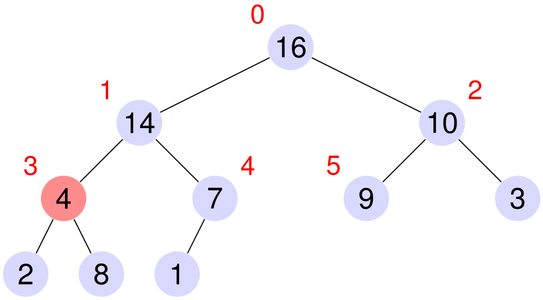
Przykład

16, 4 ,10,14,7,9,3,2,8,1



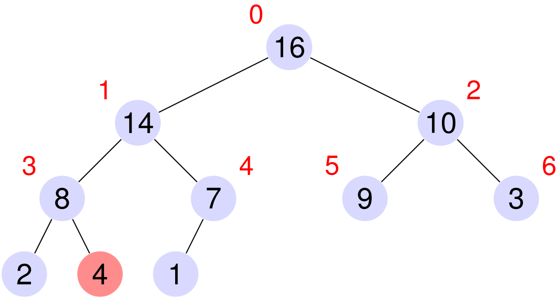
Przykład

16,14,10, 4 ,7,9,3,2,8,1



Przykład

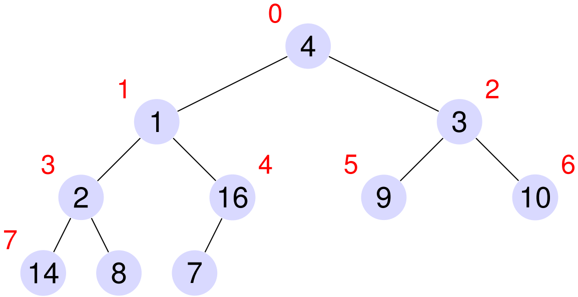
16,14,10,8,7,9,3,2, 4 ,1



```
BUILD-MAX-HEAP($A$)
heapsize = A.length // initialize heapsize
for $i = \left\lfloor \frac{A.length-2}{2} \right\rfloor$ downto 0 // 0 is the root of tree
        MAX-HEAPIFY($A$, $heapsize$, $i$)
```

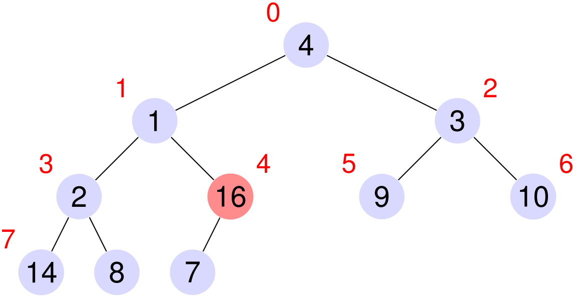
Przykład

4,1,3,2,16,9,10,14,8,7



Przykład

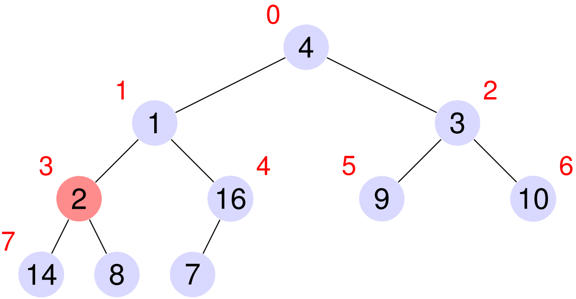
4,1,3,2, 16 ,9,10,14,8, 7



Przykład

4,1,3, 2 ,16,9,10, 14,8 ,7

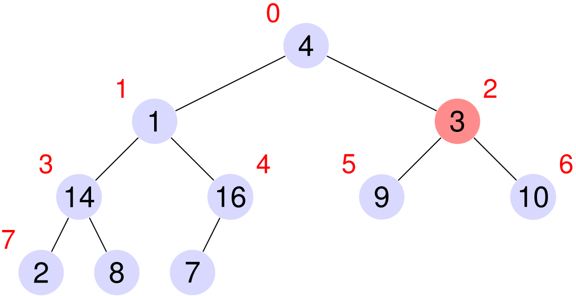
4,1,3, 14 ,16,9,10, 2,8 ,7



Przykład

4,1, 3 ,14,16, 9,10 ,2,8,7

4,1, 10 ,14,16, 9,3 ,2,8,7

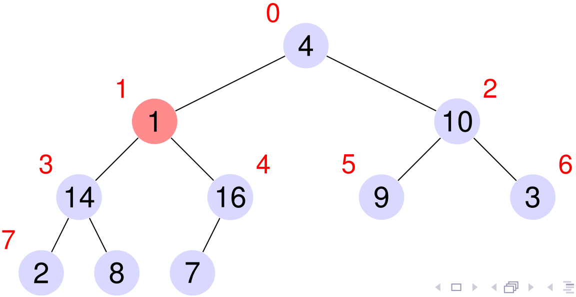


Przykład

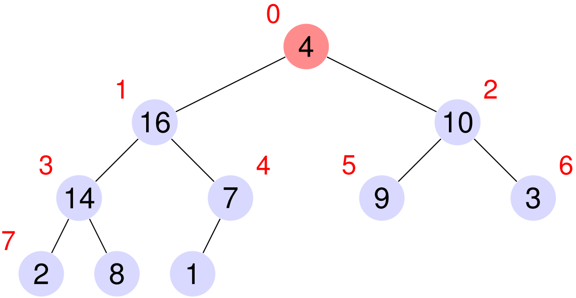
4, 1 ,10, 14,16 ,9,3,2,8,7

4, 16 ,10, 14,1 ,9,3,2,8, 7

4, 16 ,10, 14,7 ,9,3,2,8, 1



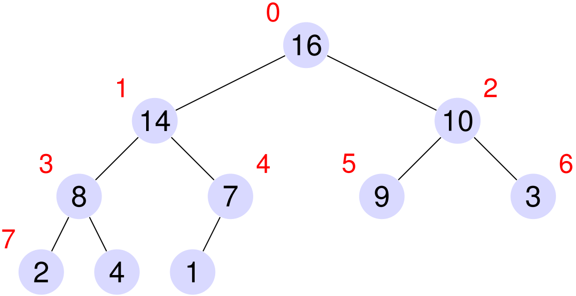
Przykład



Przykład

4,16,10,14,7,9,3,2,8,1
16,4,10,14,7,9,3,2,8,1
16,14,10,4,7,9,3,2,8,1
16,14,10,8,7,9,3,2,4,1

Przykład



```
HEAP-SORT($A$)
BUILD-MAX-HEAP($A$) // array permuted
$heapsize = len(A)$
for $i = heapsize - 1$ downto 1
    swap( $A[0]$ and $A[i]$)
    $heapsize = heapsize - 1$
    MAX-HEAPIFY($A, heapsize, 0$)
return A
```

```
DIJKSTRA($G, w, s$)
for each vertex v $\in$ V(G)
    $d[v] = \infty$
    $\pi$[v] = NIL
$d[s] = 0$
$S = \emptyset$
$Q = V(G)$
while Q$\neq \emptyset$
    u = Extract-Min(Q)
    S = $S \cup \{u\}$
    for each vertex v adjacent to u
        Relax(u,v,w)
```

```
RELAX($u, v, w$)
if $d[v] > d[u] + w(u, v)$
        $d[v] = d[u] + w(u, v)$
        $\pi[v] = u$
```

Cormen Thomas H., Leiserson Charles E., Rivest Ronald L, Clifford Stein, Wprowadzenie do algorytmów.