ALGORITHMS

UPHEAP(i, H): $\in O(\text{height})$

```
while i > 1 & H[i] > H[parent[i]] - runs most h time
    swap(i, Parent(i), H)
    i = Parent(i)
elihw
end
```

INSERT(x, H): $\in O(\text{Height})$

```
RESIZE?()
H.size += 1
H[H.size] = x
UPHEAP(H.size, H) - h
end
```

DOWNHEAP(i, H): $\in O(\text{HEIGHT})$

```
if Left(i) > H.size:
    return
fi
Li = Left(i)          - Leger child index
if RIGHT(i) ≤ H.size & H[li] < H[RIGHT(i)]
    li = RIGHT(i)
fi
if H[i] < H[li]
    swap(H, i, li)
    DOWNHEAP(li, H)
fi
end
```

REMOVE (H): $\in O(\text{HEIGHT})$

EMPTY?() → PANIC!

RESIZE?()

rV = H[1]

H[1] = H[H.size]

H.size -= 1

DOWNHEAP(1, H)

return rV

end