Heap

CS3026 – Análisis y Diseño de Sistemas

Angel Napa

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Índice

1 Heap

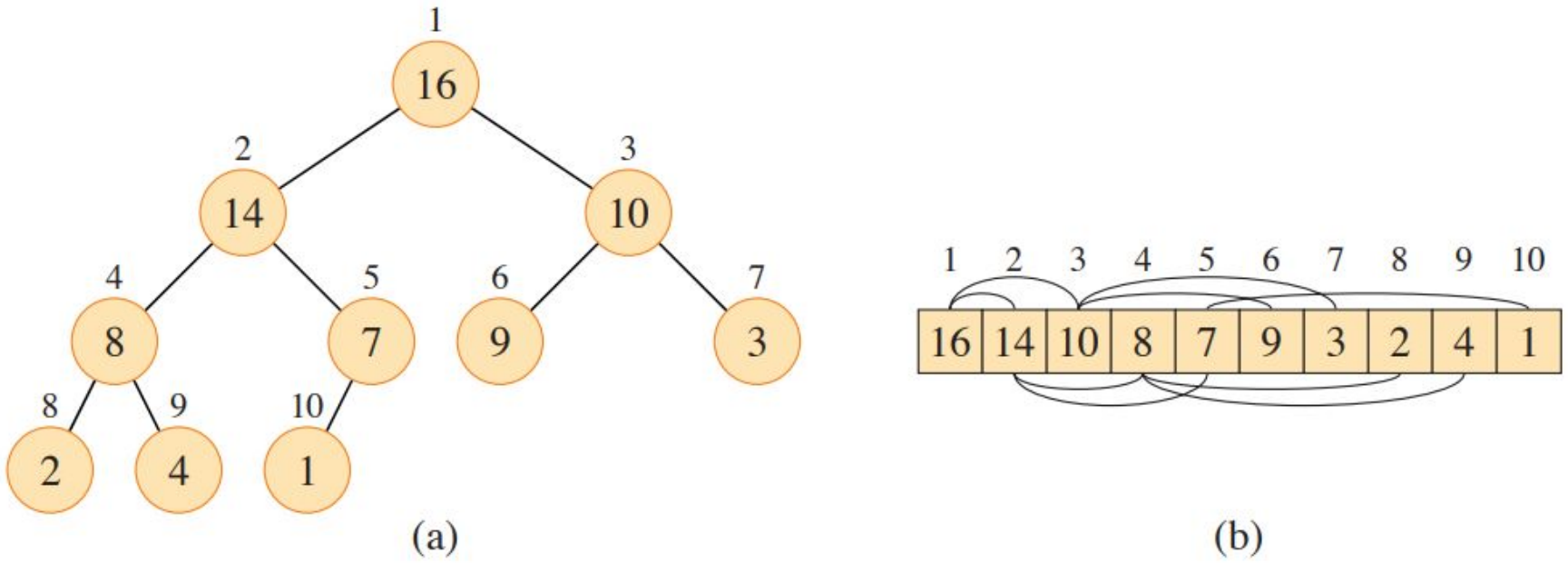

Figure 1: Cormen, Introduction to Algorithms

```
PARENT(i)
1  return ⌊i/2⌋

LEFT(i)
1  return 2i

RIGHT(i)
1  return 2i + 1
```

Figure 2: Cormen, Introduction to Algorithms

```
MAX-HEAPIFY(A, i)
 1  l = LEFT(i)
 2  r = RIGHT(i)
 3  if l ≤ A.heap-size and A[l] > A[i]
 4       largest = l
 5  else largest = i
 6  if r ≤ A.heap-size and A[r] > A[largest]
 7       largest = r
 8  if largest ≠ i
 9       exchange A[i] with A[largest]
10       MAX-HEAPIFY(A, largest)
```

Figure 3: Cormen, Introduction to Algorithms

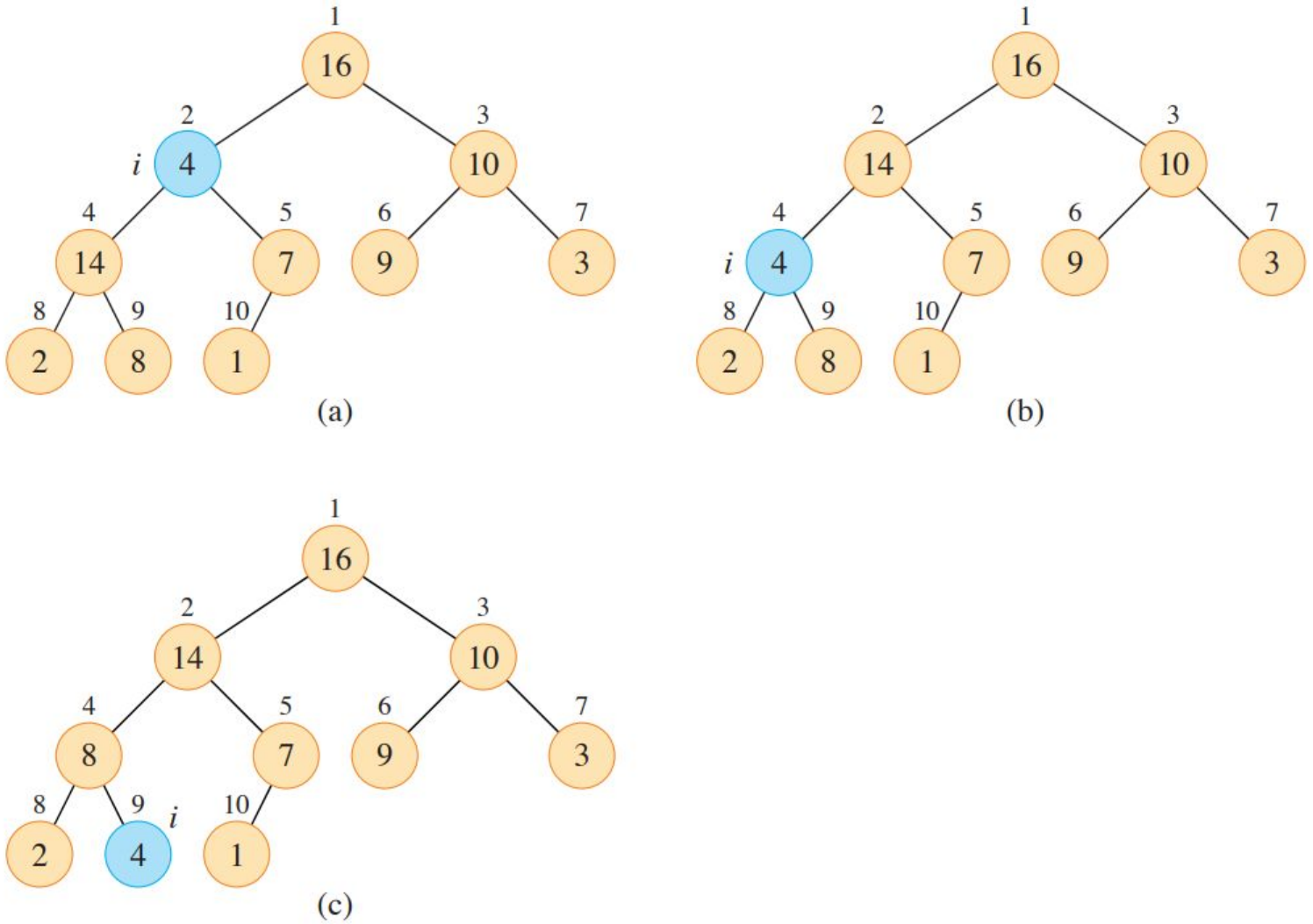

Figure 4: Cormen, Introduction to Algorithms

```
BUILD-MAX-HEAP(A, n)
1  A.heap-size = n
2  for i = ⌊n/2⌋ downto 1
3      MAX-HEAPIFY(A, i)
```

Figure 5: Cormen, Introduction to Algorithms

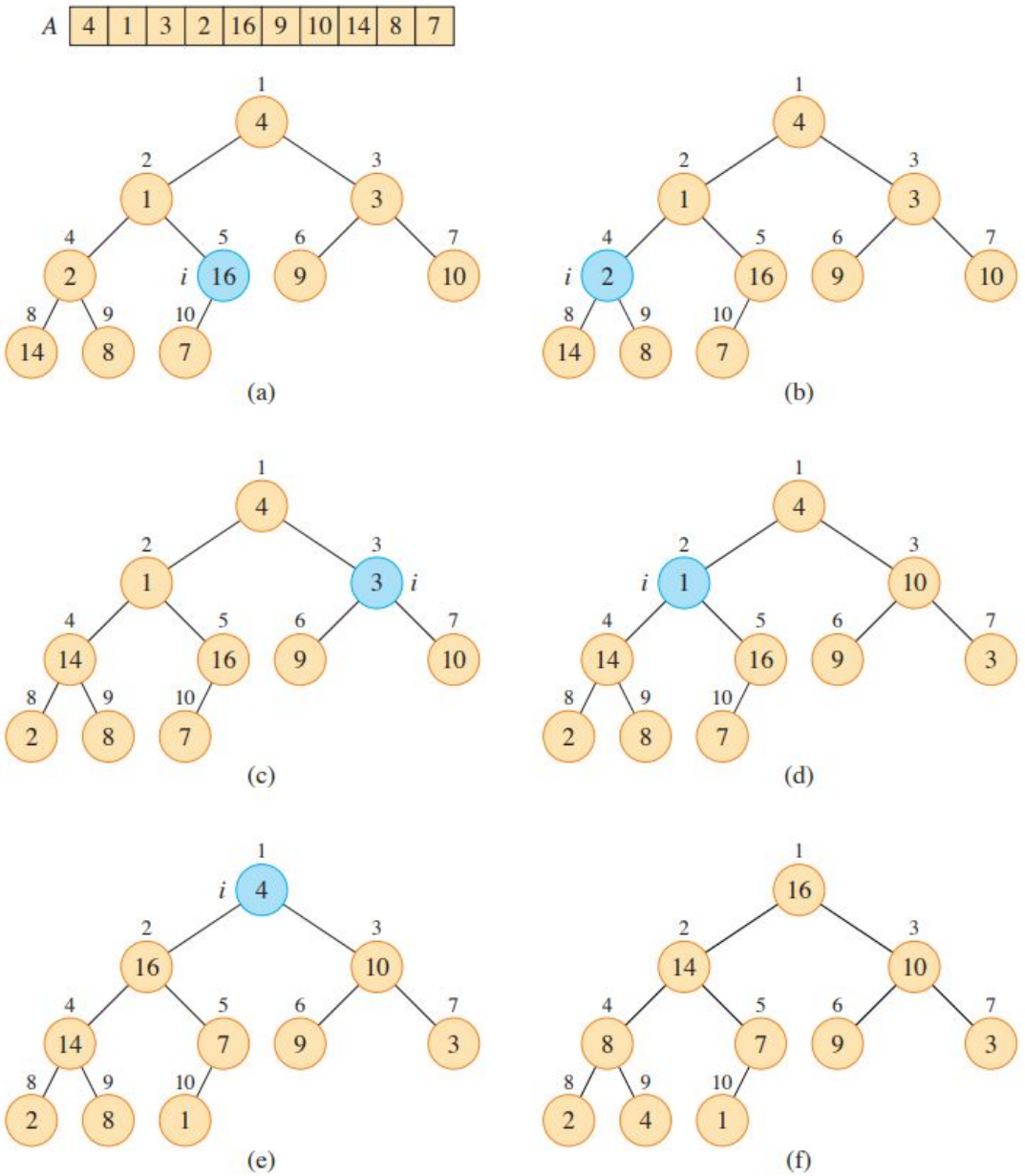

Figure 6: Cormen, Introduction to Algorithms

2 Heapsort

```
HEAPSORT(A, n)
1  BUILD-MAX-HEAP(A, n)
2  for i = n downto 2
3      exchange A[1] with A[i]
4      A.heap-size = A.heap-size − 1
5      MAX-HEAPIFY(A, 1)
```

Figure 7: Cormen, Introduction to Algorithms

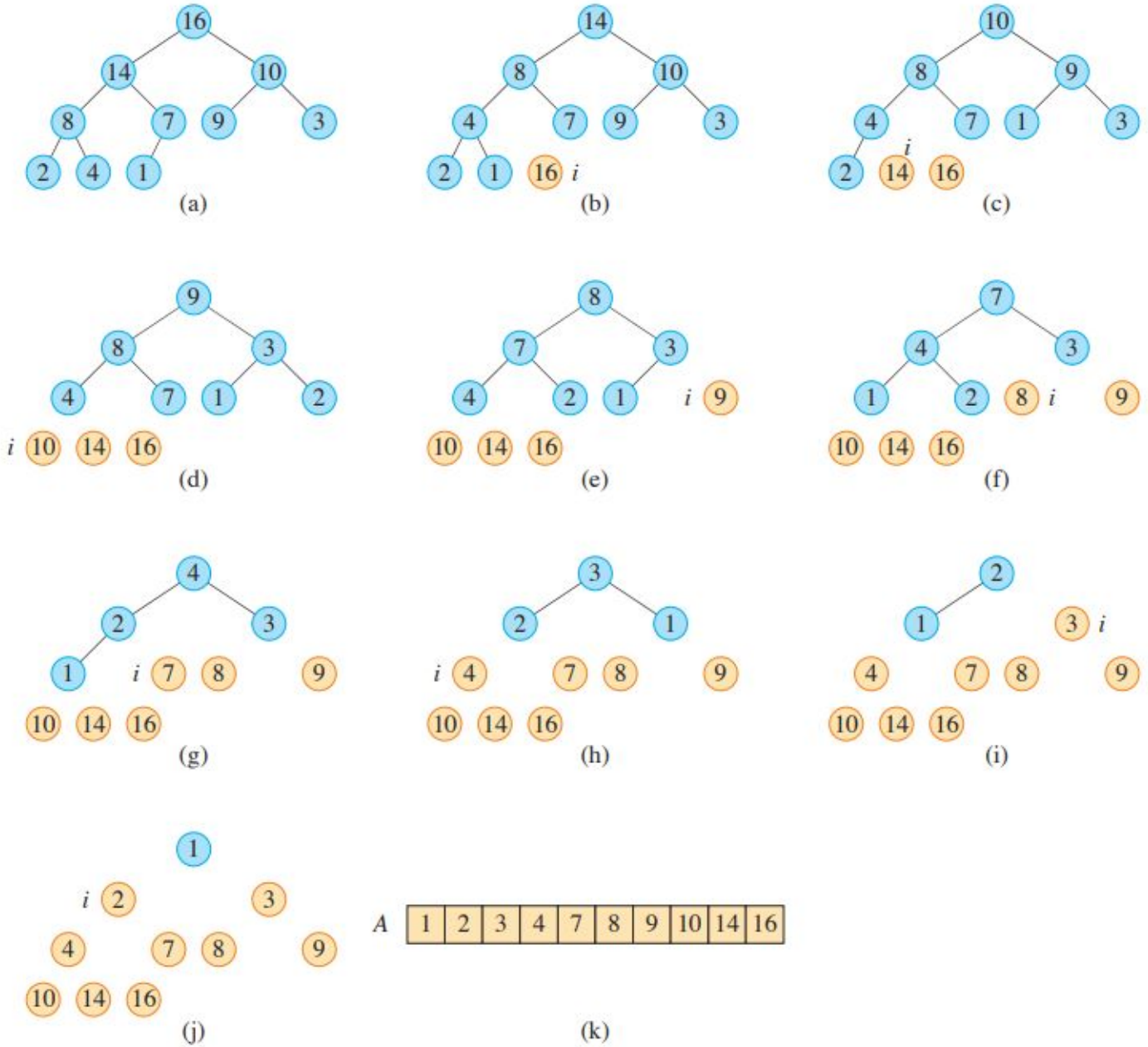

Figure 8: Cormen, Introduction to Algorithms

3 Priority Queue

```
MAX-HEAP-MAXIMUM(A)
1  if A.heap-size < 1
2        error "heap underflow"
3  return A[1]
```

Figure 9: Cormen, Introduction to Algorithms

```
MAX-HEAP-EXTRACT-MAX(A)
1  max = MAX-HEAP-MAXIMUM(A)
2  A[1] = A[A.heap-size]
3  A.heap-size = A.heap-size − 1
4  MAX-HEAPIFY(A, 1)
5  return max
```

Figure 10: Cormen, Introduction to Algorithms

```
MAX-HEAP-INCREASE-KEY(A, x, k)
1  if k < x.key
2       error "new key is smaller than current key"
3  x.key = k
4  find the index i in array A where object x occurs
5  while i > 1 and A[PARENT(i)].key < A[i].key
6       exchange A[i] with A[PARENT(i)], updating the information that maps
            priority queue objects to array indices
7       i = PARENT(i)
```

Figure 11: Cormen, Introduction to Algorithms

```
MAX-HEAP-INSERT(A, x, n)
1  if A.heap-size == n
2        error "heap overflow"
3  A.heap-size = A.heap-size + 1
4  k = x.key
5  x.key = −∞
6  A[A.heap-size] = x
7  map x to index heap-size in the array
8  MAX-HEAP-INCREASE-KEY(A, x, k)
```

Figure 12: Cormen, Introduction to Algorithms

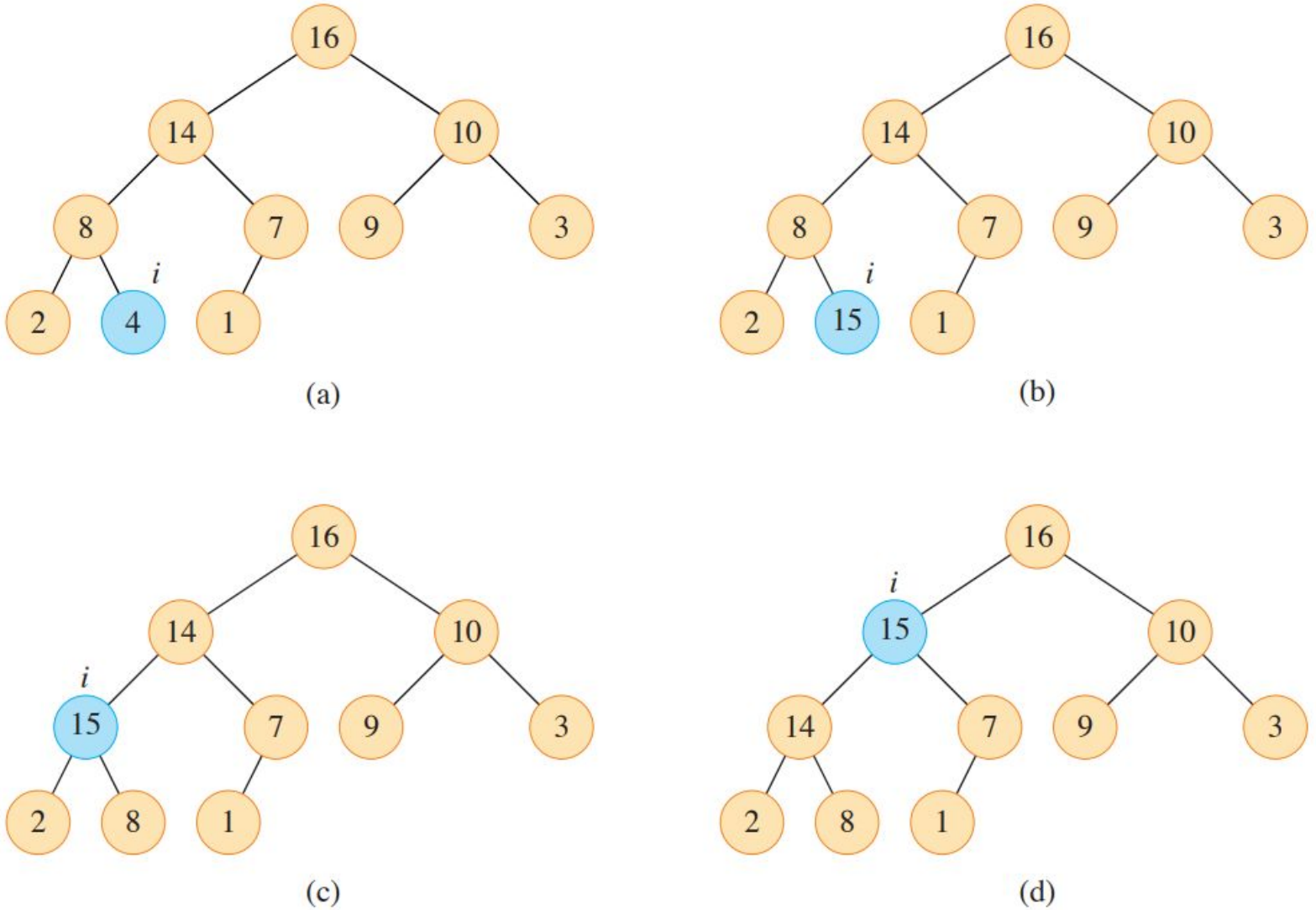

Figure 13: Cormen, Introduction to Algorithms

Gracias

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA