2-8 Probabilistic Analysis

Hengfeng Wei

hfwei@nju.edu.cn

May 09, 2020

Compactation (Problem 10.3 – 4)

Keep all elements of a doubly linked list compact in storage, using the first m index locations in the multiple-array representation.

ALLOCATE-OBJECT FREE-OBJECT

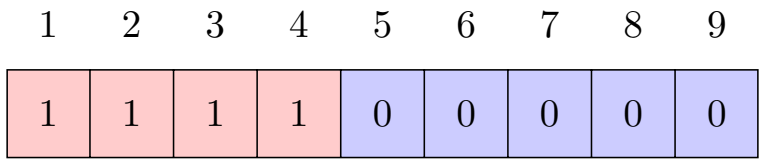
Using the array implementation of a stack.

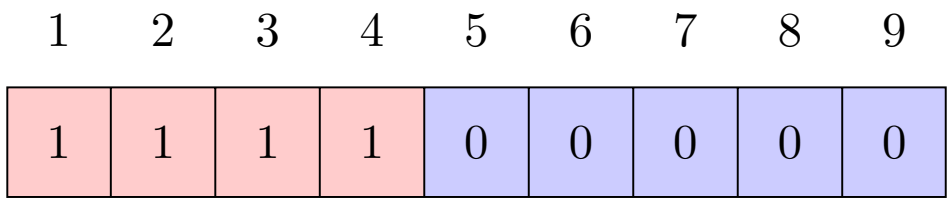
Compactify-List (Problem 10.3 – 5)

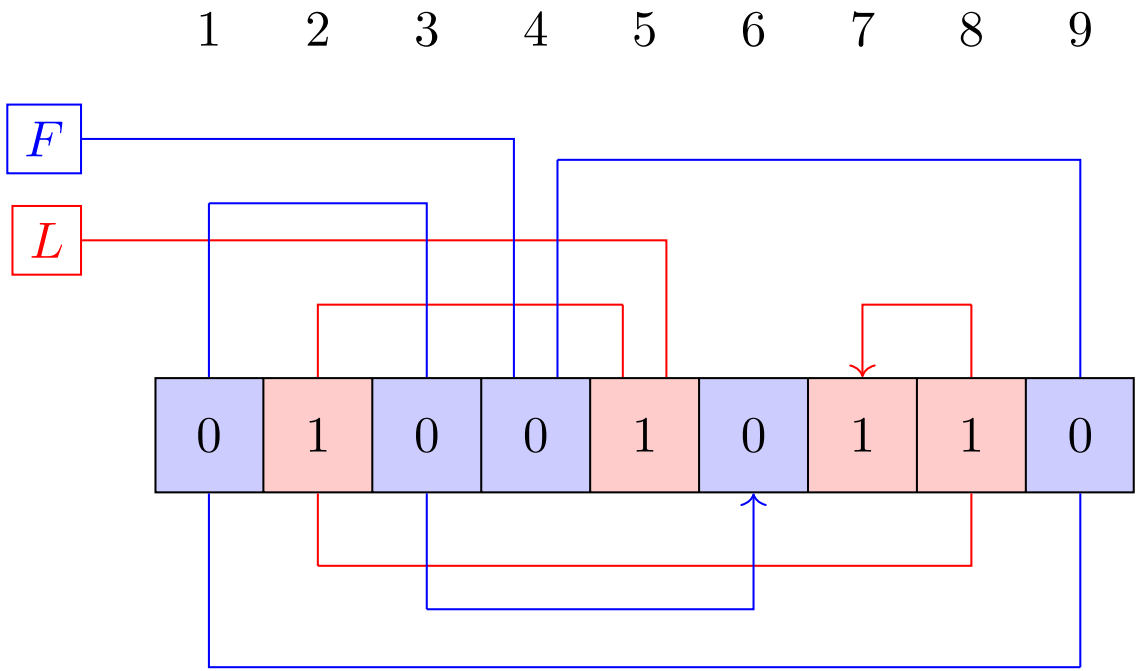
Compactify-List(L, F)

$$L: \text{ doubly linked list}, \quad |L| = n$$
$$F: \text{ doubly linked free list}, \quad |F| = m - n$$

$$\Theta(n)$$



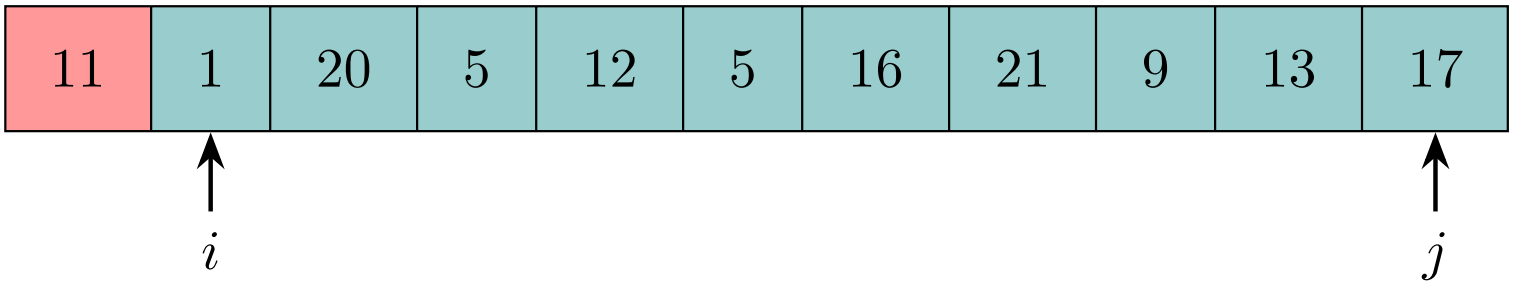




Swap $(0, 1)$ pairs

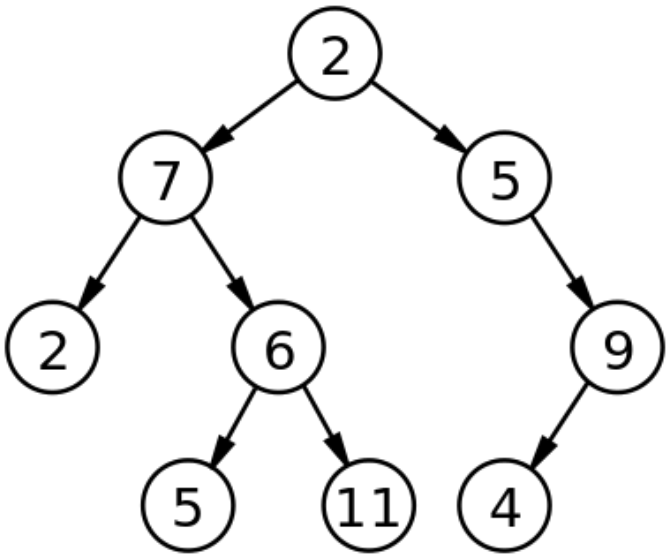
$\Theta(n)$

HOARE-PARTITION



Recursive Binary Tree Traversal (Problem 10.4 – 2)

$O(n)$



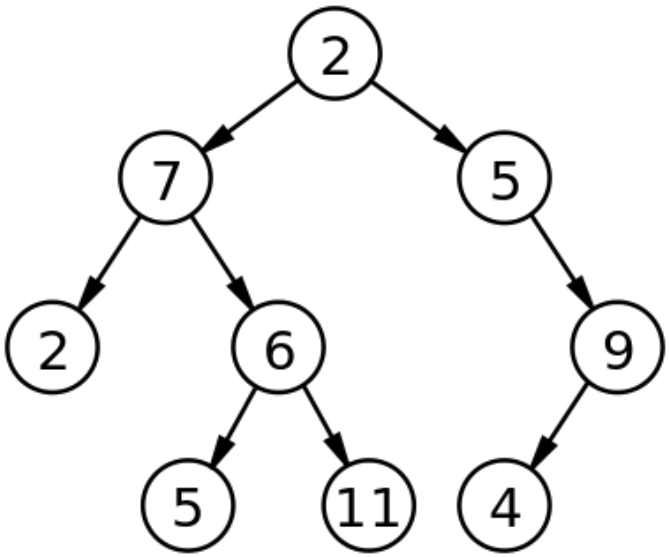
procedure RECURSIVE-DFS(t)
  print $t.key$

  if $t.left \neq$ NIL **then**
    RECURSIVE-DFS($t.left$)
  if $t.right \neq$ NIL **then**
    RECURSIVE-DFS($t.right$)

RECURSIVE-DFS($T.root$)

Non-recursive Binary Tree Traversal (Problem 10.4 – 2)

$O(n)$



procedure ITERATIVE-DFS(t)
  S.PUSH(t) ▷ S : stack

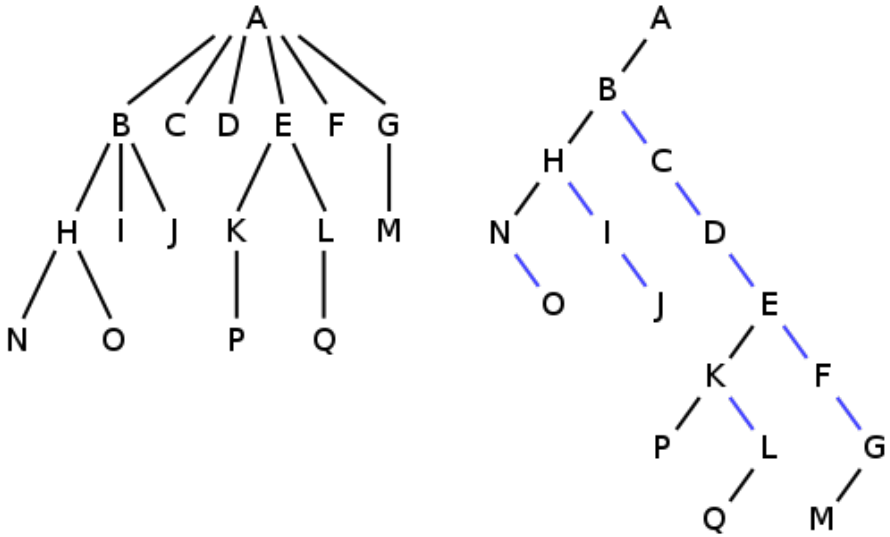
  while $S \neq \emptyset$ **do**
    $v \leftarrow S$.POP()
    print $v.key$

    if $v.right \neq$ NIL **then**
      S.PUSH($v.right$)
    if $v.left \neq$ NIL **then**
      S.PUSH($v.left$)

ITERATIVE-DFS($T.root$)

"LCRS" Tree Traversal (Problem 10.4 – 2)

$O(n)$

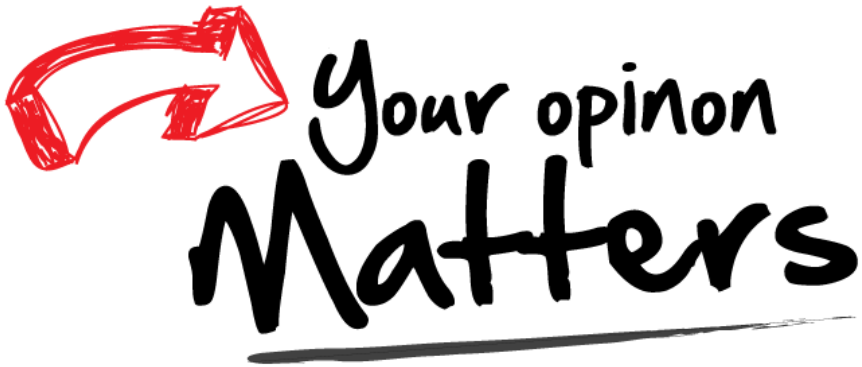


procedure RECURSIVE-DFS(t)
    print $t.key$

    if $t.lc \neq$ NIL **then**
        RECURSIVE-DFS($t.lc$)
    if $t.rs \neq$ NIL **then**
        RECURSIVE-DFS($t.rs$)

RECURSIVE-DFS($T.root$)


Thank
You!



Office 302

Mailbox: H016

hfwei@nju.edu.cn