Selected Problems in CLRS

Section 10

Notifications

Problem Difficulty (count with star)

1. you can solve w/o the brain
2. you can solve if you think a bit
3. you can solve if you think carefully
4. you might solve if you push yourself
5. you can solve if you use other's brain

Exercise

10.1-1 $\star$

Using Figure 10.1 as a model, illustrate the result of each operation in the sequence PUSH$(S,4)$, PUSH$(S,1)$, PUSH$(S,3)$, POP(S), PUSH$(S,8)$, POP(S) on an initially empty stack S stored in array $S[1..6]$.

10.1-2 $\star\star$

Explain how to implement two stacks in one array $A[1..n]$ in such a way that neither stack overflows unless the total number of elements in both stacks together is n. The PUSH and POP operations should run in $O(1)$ time.

> **Hint :** there are many ways to do it. One interesting implementation is found at linux memory management. The growth directions of stack and heap is opposite.

10.1-3 $\star$

Using Figure 10.1 as a model, illustrate the result of each operation in the sequence ENQUEUE$(Q,4)$, ENEQUEUE$(Q,1)$, ENQUEUE$(Q,3)$, DEQUEUE(S), ENQUEUE$(S,8)$, DEQUEUE(S) on an initially empty queue Q stored in array $Q[1..6]$.

10.1-6 $\star\star$

Show how to implement a queue using two stacks. Analyze the running time of the queue operatios.

10.1-7 $\star\star$

Show how to implement a stack using two queues. Analyze the running time of the stack operations.

10.2-2 $\star\star$

Implement a stack using a singly linked list L.

10.2-3 $\star\star$

Implement a queue using a singly linked list L.

10.4-1 ⋆⋆

Draw the binary tree rooted at index 6 that is represented by the following attributes

index	key	left	right
1	12	7	3
2	15	8	NIL
3	4	10	NIL
4	10	5	9
5	2	NIL	NIL
6	18	1	4
7	7	NIL	NIL
8	14	6	2
9	21	NIL	NIL
10	5	NIL	NIL

10.4-5 ⋆⋆⋆

Write an $O(n)$-time nonrecursive procedure that, given an n-node binary tree, prints out the key of each node. Use no more than constant extra space outside of the tree itself and do not modify the tree, even temporarily, during the procedure.

Note : it is important at other applications