

CS507 Computing Foundation for Computational Science HW7

Min Long

This HW is a pure written HW, and should be finished in a PDF file. To make a PDF file, you can scan your solutions, or type your solution in WORD and then convert it.

Due time: Due: 12/9/2022, 23:59

Total: 25 points

Submission command: submit minlong CS507 HW7

Instruction to electronic submission: <http://cs.boisestate.edu/~cs221/SubmissionProcedure.html>

- The written assignment can be done in a pure text format (*.txt, for example, problem1.txt) on Onyx.
- The programming assignment should be presented with source codes.
- Each problem should have its own working directory, such as HW2/prob1, HW2/prob2 ... For example, the following table shows the structure of HW1 from the user “student1” and how to submit HW to us through Onyx

```
1 [student1@onyx:HW1]$ ls -l
2 drwxr-x---. 2 student1 Students 13 Sep 19 2021 prob1
3 drwxr-xr-x. 3 student1 Students 14 Sep 19 2021 prob2
4 [student1@onyx:HW1]$ cd prob1/
5 [minlong@onyx:prob1]$ ls
6 -rw-r-----. 1 student1 Students 943 Sep 19 2021 problem1.txt
7 $ cd ..
8 [student1@onyx:HW1]$ pwd
9 /home/student1/CS507/HW1
10 [student1@onyx:HW1]$ submit minlong CS507 HW1
```

Listing 1: A sample structure of homework and submission procedure.

- Your source codes (if any) must compile and run on Onyx.
- Documentation is important and proper comments are expected in your source code.
 - comments giving description of: purpose, parameters, and return value if applicable
 - other comments where clarification of source code is needed
 - proper and consistent indentation
 - proper structure and modularity

Don't ask us or your classmates directly for solutions (it happened); just try as much as possible. Be patient and enjoy coding!

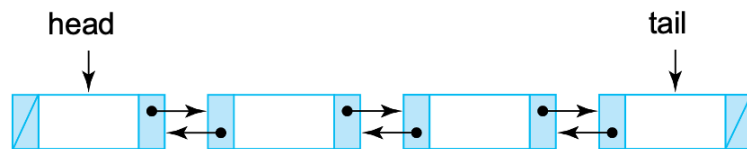
Written Problems

1. (5 points) (Stack and Queue). There are many ways to implement a stack, such as using an array, a linked list, and even using queues. How to use two queues to implement a stack so that push runs in $O(1)$ and pop runs in $O(n)$? Suppose the queues have no size limit. Please describe your algorithm without pseudocode.

Hint: Queue—First In, First Out; Stack—Last In, First Out.

2. (10 points) (Doubly Linked List) Please write a pseudocode for `List-move(x, y)`. This procedure is to move an existing node x to the front of another existing node y in a doubly non-circular list (i.e., a regular doubly linked list with a head and tail).

Hint: “Existing” means both x and y are already in the list. So there should be 2 steps: (1) detach x , (2) insert x . In addition, you need to think carefully about different cases when x and y are in some special positions in the list.



A pseudocode to move existing x to the front of existing y
`List-move(L, node x, node y)`

#detach x

#insert x

3. (10 points) (QuickSort) For a given input array $A : < 6, 4, 7, 9, 1, 8, 2, 3, 5 >$, what is the sequence of numbers in A after the first partition (by calling `Partition(A, 1, 9)`)? Note that 1 and 9 in `Partition(A, 1, 9)` function call are array indices. Please show the intermediate steps. (`Partition(A, p, r)` is part of QuickSort and its pseudocode can be found in P. 171 in CLRS or as follows.)

Algorithm 1 `Partition(A,p,r)`

```
1:  $x = A[r]$ 
2:  $i = p - 1$ 
3: for  $j \leftarrow p, r - 1$  do
4:   if  $A[j] \leq x$  then
5:      $i \leftarrow i + 1$ 
6:      $A[i] \leftrightarrow A[j]$ 
7:   end if
8: end for
9:  $A[i + 1] \leftrightarrow A[r]$ 
10: return  $i + 1$ 
```

6, 4, 7, 9, 1, 8, 2, 3, 5 \\ Now start sorting