CS240: Data Structures
Winter Semester – Academic Year 2017-18
Panagiota Fatourou

2nd Set of Theoretical Exercises

Submission deadline: Monday, 30 October 2017
How to submit: The assignments must be submitted either in electronic format using the turnin program (see instructions on the website of the course), or to the teaching assistants of the course in the laboratory of postgraduates, on Monday, 30 October 2017, time 15:00-16:00. Assignments submitted after 16:00 of Monday, 30/10/2017 are considered out of date. Out of date assignments are accepted only in electronic format using the turnin program.

Exercise 1 [40 points]

Let a library provide you access to stacks and queues of characters. The library allow you to set a stack(or a queue) and call 5 basic functions in it. For example, the definition of a stack(or a queue) happens by writing: Stack S; (respectively, Queue Q;). The following operations are supported for the stack: (1) void MakeEmptyStack(Stack S), (2) boolean IsEmptyStack(Stack S), (3) Type Top(Stack S), (4) Type Pop(Stack S), (5) void Push(Stack S, char x). Respectively, the following operations are supported for the queue: (1) void MakeEmptyQueue(Queue Q), (2) boolean IsEmptyQueue(Queue Q), (3) char Front(Queue Q), (4) char Dequeue(Queue Q), (5) void Enqueue(Queue Q, char x).

Solve the problems listed below:

a. In post-order expression, every operation is applied to the two operatives which follow it. For example, '1 2 +' represents the sum of the number 1 with the number 2. The definition is recursive and so every operation may refer to operatives that are the result of other operations. E.g., '1 2 + 3 4 – * 5 -' is the post-order representation of '(1+2)*(3-4)-5'. Present an algorithm which will read an expression in post-order form and it will return the result of the calculation of the expression. [15P]

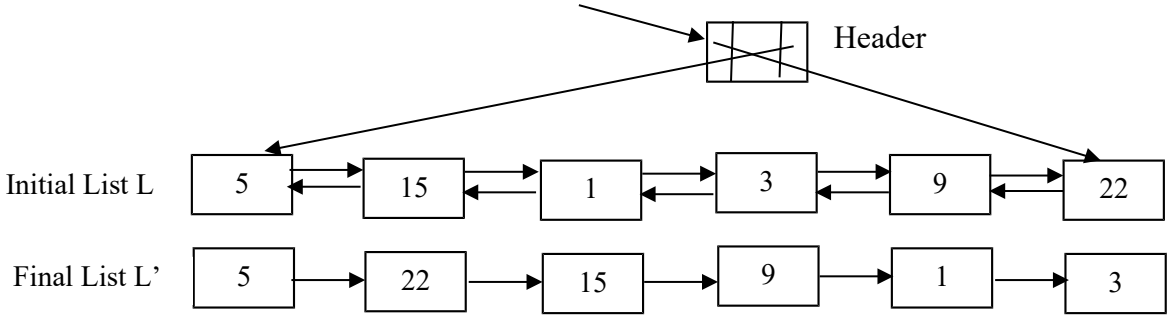
b. Present an algorithm which will read n elements from the keyboard and it will put them on a stack S_1. Subsequently, your algorithm should classify the elements of S_1, using only one additional auxiliary stack(and auxiliary variables, e.g integers). The classification should be performed by following the InsertionSort() algorithm technique. After the end of the algorithm execution, the deepest element of S_1 should be the smallest element of what was read, while the top element of S_1 should be the largest element. [25P]

Notes: I. The algorithms for both queries should use queues and stacks from the library that is provided to you, as well as possibly some auxiliary variables. It is not allowed to use tables or other non-library data structures for storing or processing alphanumeric or character sequences.

II.In query (a), consider that you are provided with an auxilary routine ch = getchar () for reading numbers or characters. In addition, you are provided with the IsOperand (char ch) and IsOperator (char ch) auxilary functions, which return TRUE if ch is an operand and an operator, respectively. Also consider that all the numbers of each transformed representation are single digits.

Exercise 2 [25 points]

Present an algorithm which given a **double-linked list** L with n elements, will create another **simple-linked list** L' with also n elements for which the following should apply. The elements of L' that are in odd positions (1st, 3rd, 5th, etc.) are copies of the first n / 2 elements of L, while the elements of L' that are in even positions (2nd, 4th, 6th, etc.) are copies of the last n / 2 elements of L (from the end to the beginning). n can be any even number (eg, 0, 2, 4, etc.). Your algorithm should run at time O(n). Assume that the double-linked list is circular (as in the slides) and that a pointer H on the Header node is given. The nodes of the list are consisted of the fields num, prev and next.



The above figure presents an example of a list L and the list L' which should be created. Consider that n is not known in the algorithm (that is if you need it, the algorithm you will plan should compute it).

Exercise 3 [35 points]

a. An array A with n rows and n columns is called tridiagonal if for every i,j, $0 \le i,j \le n-1$, is $A[i,j] = 0$, if $i \ne j$ and $i \ne j-1$ and $i \ne j+1$. An example of tridiagonal array is presented in figure 3. Describe a data structure for storing of a tridiagonal array which will use space O(n). Present pseudocode for the function Access(A,i,j), which returns the value of the element A[i,j]. What is the complexity of Access () you have implemented? [20P]

12	11	0	0	0
34	43	44	0	0
0	76	23	31	0
0	0	7	62	61
0	0	0	1	32

Figure 3

b. Consider a special ladder linking the Earth to some space stations and let one of the space sports of the future be the conquest of the ladder stairs(that is, that an athlete to ascends as many stairs as possible starting from the Earth, of course carrying the appropriate space equipment that will allow him to survive in the conditions he encounters in each phase of the stair climb). With the passage of time, the number of stairs that the athletes have conquered is increasing (that is, periodically athletes make new space records climbing more and more stairs). However, the ladder is no-ending.

The ladder has at each stair a special slot where each athlete leaves a small flag which indicates that the athlete has conquered this stair of the ladder. Stairs that have no flag haven't yet be conquered, while those that have been conquered are flagged. Present an algorithm, which will find the total number, n, of the stairs that have been conquered. Your algorithm should have time complexity O (logn). [15P]