

Elementary Data Structures and Algorithms

Stacks

Concept: *input-output order*

1. These values are pushed onto a stack in the order given: 1 5 9. A *pop* operation would return which value?
 - A. 1
 - B. 5
 - C. 9
2. LIFO ordering is the same as:
 - A. FIFO
 - B. FILO
 - C. LILO

Concept: *time and space complexity*

3. Consider a stack based upon a fillable array with pushes onto the back of the array. What is the time complexity of the worst case behavior for *push* and *pop*, respectively? You may assume there is sufficient space for the *push* operation.
 - A. linear and linear
 - B. linear and constant
 - C. constant and constant
 - D. constant and linear
4. Consider a stack based upon a circular array with pushes onto the front of the array. What is the time complexity of the worst case behavior for *push* and *pop*, respectively? You may assume there is sufficient space for the *push* operation.
 - A. constant and constant
 - B. constant and linear
 - C. linear and linear
 - D. linear and constant
5. Consider a stack based upon a dynamic array with pushes onto the back of the array. What is the time complexity of the worst case behavior for *push* and *pop*, respectively? You may assume there is sufficient space for the *push* operation and that the array never shrinks.
 - A. linear and linear
 - B. constant and linear
 - C. constant and constant
 - D. linear and constant

6. Consider a stack based upon a dynamic array with pushes onto the back of the array. What is the time complexity of the worst case behavior for *push* and *pop*, respectively? You may assume there is sufficient space for the *push* operation and that the array may shrink.
- A. constant and constant
 - B. linear and linear
 - C. linear and constant
 - D. constant and linear
7. Consider a stack based upon a dynamic circular array with pushes onto the front of the array. What is the time complexity of the worst case behavior for *push* and *pop*, respectively? You may assume the array may grow or shrink.
- A. constant and constant
 - B. linear and linear
 - C. linear and constant
 - D. constant and linear
8. Consider a stack based upon a singly-linked list without a tail pointer with pushes onto the front of the list. What is the time complexity of the worst case behavior for *push* and *pop*, respectively?
- A. linear and linear
 - B. constant and linear
 - C. constant and constant
 - D. linear and constant
9. Consider a stack based upon a singly-linked list with a tail pointer with pushes onto the front of the list. What is the time complexity of the worst case behavior for *push* and *pop*, respectively?
- A. linear and constant
 - B. linear and linear
 - C. constant and linear
 - D. constant and constant
10. Consider a stack based upon a non-circular, doubly-linked list without a tail pointer with pushes onto the front of the list. What is the time complexity of the worst case behavior for *push* and *pop*, respectively?
- A. linear and linear
 - B. constant and constant
 - C. constant and linear
 - D. linear and constant
11. Consider a stack based upon a doubly-linked list with a tail pointer with pushes onto the front of the list. What is the time complexity of the worst case behavior for *push* and *pop*, respectively?
- A. linear and constant
 - B. linear and linear
 - C. constant and linear
 - D. constant and constant

12. Suppose a simple fillable array with capacity c is used to implement two stacks, one growing from each end. The stack sizes at any given time are stored in i and j , respectively. If maximum space efficiency is desired, a reliable condition for the stacks being full is:

- A. $i == c/2 \ || \ j == c/2$
- B. $i + j == c$
- C. $i + j == c-2$
- D. $i == c/2-1 \ || \ j == c/2-1$
- E. $i == c/2-1 \ \&\& \ j == c/2-1$
- F. $i == c/2 \ \&\& \ j == c/2$

Concept: *stack applications*

For the following questions, assume the tokens in a post-fix equation are processed with the following code, with all functions having their obvious meanings and integer division.

```
s.push(readEquationToken());
s.push(readEquationToken());
while (moreEquationTokens())
{
    t = readEquationToken();
    if (isNumber(t))
        s.push(t);
    else /* t must be an operator */
    {
        operandB = s.pop();
        operandA = s.pop();
        result = performOperation(t,operandA,operandB);
        s.push(result);
    }
}
```

13. If the tokens of the postfix equation $8 \ 2 \ 3 \ ^ \ / \ 2 \ 3 \ * \ + \ 5 \ 1 \ * \ -$ are read in the order given, what are the top two values in s immediately after the result of the first multiplication is pushed?

- A. 1 6
- B. 3 3
- C. 1 2
- D. 5 6