

Due Monday, January 23<sup>rd</sup>, 4:00 pm in 2131 Kemper

1. (2 points) Let  $A$  be an array of size  $n \geq 6$  containing integers from 1 to  $n - 5$ , inclusive with exactly five repeated. Describe an  $O(n)$  algorithm for finding the five integers in  $A$  that are repeated. (Goodrich, p. 150)
2. (2 points) Given a list  $L$  of  $n$  arbitrary integers, describe an  $O(n)$  time function for finding an integer that cannot be formed as the sum of two integers in  $L$ . (Goodrich, p. 266)
3. (2 points) Describe how to implement a stack ADT using two queues. What is the running time of the `push()`, and `pop()` function in this case? (Goodrich p. 224)
4. (2 points) Propose a data structure that supports the stack push and pop operations, and a third operation `findMin`, which returns the smallest element in the data structure, all in  $O(1)$  worst-case time. (Weiss, p. 119)
5. (2 points) If the recursive routine on page 59 of the text used to compute Fibonacci numbers is run for  $N = 50$ , is stack space likely to run out? Why or why not? (Weiss p. 119)

Sources of questions:

Michael T. Goodrich, Roberto Tamassia, and David Mount, *Data Structures & Algorithms, Second Edition*, Hoboken, NJ, John Wiley & Sons, 2011.

Mark Weiss, *Data Structures and Algorithm Analysis in C++, Fourth Edition*, New York, NY, Pearson Education, 2014.