# 2002 Sun Microsystems and TopCoder Collegiate Challenge – Problem Statement

MaxDensity PROBLEM STATEMENT

Given a set of non-negative integers S, and a non-negative integer N, find a closed interval of length N that starts on a non-negative number and contains the largest number of elements of S. Return the first element of that interval. If there are multiple such intervals, choose the one that starts with the lowest non-negative number.
Closed interval is defined as follows: given two integers a and b such that a<=b, closed interval [a,b] ::= all integer x such that a<=x<=b; the length of the interval equals b-a. For example, [5,7] has the length of 2; [100,100] has the length of 0.

DEFINITION
Class name: MaxDensity
Method name: getMaxDens
Parameters: int[], int
Return type: int
Method signature (be sure your method is public): int getMaxDens (int[] S, int N);

S specifies the numbers in the set; N specifies the length of the interval. Your method should return the beginning of the first interval of length N that starts on a non-negative number, and encloses the maximum number of elements of S.

TopCoder will ensure the validity of the inputs. Inputs are valid if all of the following criteria are met:
- S has between 1 and 50 elements, inclusive,
- S contains values from 0 to 2,000,000,000, inclusive,
- S contains no duplicate values,
- N is in the range from 0 to 2,000,000,000, inclusive.

NOTE
While calculating the result, ignore all intervals that start with a negative number (see example 2).

EXAMPLES

1. S={1, 2, 3, 100, 101, 102, 103, 200, 205}, N=5. There are three closed intervals of length 5, each containing 4 elements of S: [98,103], [99,104], and [100,105]. Your method should return 98, since it is the beginning of the interval that starts with the lowest non-negative number.

2. S={0,1,2}, N=3. There are two closed intervals of length 3 that contain all three elements of S: [-1,2] and [0,3]. Since the first interval starts with a negative number, your method should return 0, which is the beginning of the second interval.

3. S={1,2,4,10,11,12}, N=2. Your method should return 10.

4. S={1,100,2000000,1999999}, N=100. Your method should return 0.

5. S={1,100,2000000,1999999,1999998}, N=100. Your method should return 1999900.

6. S={0,1000000001,2000000000}, N=1000000000. Your method should return 1000000000.

7. S={1000}, N=0. Your method should return 1000.