Homework Assignment #5
Due: February 24, 2020 at 2:30 p.m.

1. Let $A[1..n]$ be an array of integers from the range $[-n^3, n^3]$, where $n \geq 2$. Write an algorithm that, given A and an integer t, determines whether any two of the elements of A sum to t. (If $t/2$ occurs twice in A, then that counts as a pair, but if $t/2$ occurs only once in A, that does not count as a pair that sums to t.) Your algorithm should run in $O(n)$ time and use $O(n)$ space in the worst case.

You do not have to give a detailed proof of correctness, but you should state loop invariants for any loop that you use.

2. On Assignment 2, you looked at an interpolation search algorithm that searches for a value in a sorted array of n elements. You proved that the algorithm's worst-case time was $\Theta(n)$. However, it can be shown that if the numbers stored in the array are randomly and uniformly distributed, then the average running time is $O(\log \log n)$. Use this fact to briefly describe a search algorithm whose average running time is $O(\log \log n)$ when inputs are randomly and uniformly distributed, and whose worst-case running time is $O(\log n)$.

Hint: Your answer should be *very* short.