CS 161: Recitation 2 (Fall 2016)

Question 1

Given two arrays A and B of size n and a number k, determine whether there is a pair of indices (i, j) such that $A[i] + B[j] = k$.

Question 2

Suppose you have k sorted arrays, each with n elements, and you want to combine them into a single sorted array of kn elements.

(a) Recall the MERGE method used in MERGESORT. One strategy is to merge the first two arrays, then merge in the third, then merge in the fourth, and so on. What is the time complexity of this algorithm in terms of k and n?

(b) Propose a more efficient solution to this problem, using divide-and-conquer.

Question 3

Solve the recurrences below giving tight upper bounds of the form $T(n) = O(f(n))$ for an appropriate function f. You can use any method from class. Show your work. If you wish, you may assume that n initially has the form $n = a^i$, for an appropriate constant a.
Note: Unless stated otherwise, log refers to log base 2, and ln refers to the natural logarithm.

(a) $T(n) = 9T(\frac{n}{3}) + n^2$

(b) $T(n) = 4T(\sqrt{n}) + \log n$

(c) $T(n) = T(\frac{n}{3}) + T(\frac{n}{6}) + n$

Hint: In some of the recurrences, you can write another recurrence $S(m) = T(f(n))$ for some function f.