Algorithms

1. Heap data type supports the operations `INSERT` and `DELETE-MIN`. The operation `INSERT`(H, x) inserts the element x into the heap H and `DELETE-MIN`(H) deletes and returns the minimum element from H.

   (a) How is a heap implemented efficiently using an array. What is the worst-case time complexity of the operations `INSERT` and `DELETE-MIN` in this implementation ?

   10 pts

   (b) Is it possible to devise a data structure where the worst-case time complexity of both the operations, `INSERT` and `DELETE-MIN`, is $O(1)$? Provide a proof for your answer.

   20 pts

   (c) Is it possible to devise a data structure where the **amortized** time-complexity of both `INSERT` and `DELETE-MIN` is $O(\lg\lg n)$? Provide a proof for your answer.

   10 pts

2. Consider the following problem: Given a set A of n positive integers $A = \{a_1, a_2, \ldots a_n\}$, with each $a_i \leq 10n^2$, determine if there exists a subset S of A such that the sum of the elements in S is the same as the sum of the elements in $A - S$ ($\Sigma_{a_i \in S}\ a_i = \Sigma_{a_i \in A-S}\ a_i$). (**Note:** Please read both parts (a) and (b) before you write down your answers.)

   (a) Provide a polynomial time algorithm to solve the above problem. Analyse the worst-case time and space complexity of your algorithm.

   40 pts

   (b) Provide a polynomial time algorithm to solve the problem in $O(n^3)$ space. **Note:** If the space complexity of the algorithm provided for part (a) is $O(n^3)$ you can just state that here without repeating the algorithm.

   20 pts