CS2223
HW#6

DUE: Monday , December 8

1. (6 points) Suppose that given an integer constant n and an array declared int $T[n]$, you know that T is *partially sorted* in the sense that there exists an integer m , $0 \leq m \leq n-1$, such that $T[0] \leq T[1] \leq ... \leq T[m]$ and $T[m+1] \leq T[m+2] \leq ... \leq T[n-1]$. However, you do not know the value of m. Show that T can be sorted in worst-case time $O(n)$. A precise pseudo-code description of your algorithm is acceptable.

2. (5 points) A hidden memory consumption of *QuickSort* is the system stack. That is, each nontrivial invocation (invocation for a list of length at least two) of *quicksort* causes two new invocations. While the first of these new invocations is being processed, the a set of parameters are placed in the system stack.

   (a) How large can the ststem stack get?

   (b) Describe a modification of *quicksort* which would limit the worst-case size of the system stack to $O(\lg n)$, where n is the number of elements in the list to be *quicksort*ed.

3. (6 points) A typical dynamic programming algorithm provides the cost of a solution or establishes the existence of a solution without actually constructing the solution. To see how to construct a solution by using an efficient mechanism which tests for the existence of a solution, solve the following:

You are given a boolean function *BlackBox* of two inputs:

-a list of integers $x_1, x_2, ..., x_n$,

-an integer q,

and you are told that, in time $O(1)$, *BlackBox* will return "true" if there is some subset of $x_1, x_2, ..., x_n$ whose sum is q and *false* otherwise. Design an algorithm (a program is not needed) with the same input which will return an actual subset of $x_1, x_2, ..., x_n$ whose sum is q, if such a subset exists, or it should return "failure".

For example, *BlackBox*((23,27,41,72,-4,6), 29) would return "true", but your algorithm with the same input would return (27,-4,6). Your algorithm may call *BlackBox* as often as it wishes and it should work in time $O(n)$.