CSE 3500 – Algorithms and Complexity
Homework 7

Question 1 (30 points)

Consider a sorted array A of n distinct integers. Provide an algorithm to identify an index i where $1 \leq i \leq n$ and $A[i] = i$, given that such an index exists. If multiple such indices exist, the algorithm can return any one of them. Use a divide and conquer approach to solve the problem. Ensure that your algorithm achieves a time complexity better than $O(n)$, where n represents the number of integers in the array.

a. Write a clear divide and conquer algorithm to solve the problem.
b. Provide the recurrence relation that describes the time complexity of your algorithm.
c. Analyze the time complexity of your algorithm. If applicable, use the Master Theorem.

Question 2 (70 points)

The objective of this assignment is to implement Quicksort with two different pivot selection strategies and analyze their performance on various input sizes and types.

Tasks:

1. Implement Quicksort in your preferred programming language (e.g., Python) with the following pivot selection strategies:
   a. Pick the last element as the pivot.
   b. Pick a random element from the list as the pivot.
2. Test both implementations of Quicksort on different types of inputs (Sorted list, Reverse sorted list, Random list) and on different sizes (100, 500, 1000, 1500, and 2000) and measure the running time of each Quicksort implementation for each input size and type.

Submission:

1. (50 points) Submit a report that includes:
   a. A table showing the running time for each pivot selection strategy, input size, and input type.
   b. A conclusion based on the results obtained from the experiments.
2. (20 points) Submit your implementation code.

Note: Python uses a maximum recursion depth of 1000 to ensure no stack overflow errors and infinite recursions are possible. You can change the maximum recursion depth in Python.

To do this, call the sys.setrecursionlimit() function.

For example, let's set the maximum recursion depth to 3000:

```
import sys
print(sys.setrecursionlimit(3000))
```