CS2223
HW#3

DUE: Thursday, November 15

1 (8 points) \( a \) Let \( T \) be an unsorted array of \( n \) integers. Give an algorithm to find a pair \( x, y \in T \) which maximizes \( |x - y| \). If \( T = (6,13,19,3,8) \), then \( x=19 \) and \( y=3 \) would be a solution. The worst case execution time of your algorithm must be in \( O(n) \).

\( b \) Let \( T \) be a sorted array of \( n \) integers. Give an algorithm to find a pair \( x, y \in T \) which maximizes \( |x - y| \). If \( T = (3,6,8,13,19) \), then \( x=3 \) and \( y=19 \) would be a solution. The worst case execution time of your algorithm must be in \( O(1) \).

\( c \) Let \( T \) be an unsorted array of \( n \) integers. Give an algorithm to find a number \( x \) which doesn’t appear in \( T \). If \( T = (6,13,19,3,8) \), then \( x=5 \) be a solution. The worst case execution time of your algorithm must be in \( O(n) \).

\( d \) Let \( T \) be a sorted array of \( n \) integers. Give an algorithm to find a pair \( x, y \in T \) which minimizes \( |x - y| \). If \( T = (3,6,8,13,19) \), then \( x=6 \) and \( y=8 \) would be a solution. The worst case execution time of your algorithm must be in \( O(n) \).

2. (4 points) From Baase and Van Gelder’s Computer Algorithms

Suppose an algorithm does \( m^2 \) steps on an array of \( m \) elements (for any \( m \geq 1 \)). The algorithm is to be used on two arrays \( A_1 \) and \( A_2 \) (separately). The arrays contain a total of \( n \) elements. \( A_1 \) has \( k \) elements and \( A_2 \) has \( n-k \) elements (\( 0 \leq k \leq n \)).

For what value(s) of \( k \) will the most work be done? For what value(s) of \( k \) will the least work be done? Justify your answers. (Remember that an example is not a proof. There is a good solution for this problem using simple calculus.)

3. (16 points) Write programs to implement a (min)-priority queue using each of the following data structures:

- Ordered array
- Unordered array
- Binary search tree
- Heap

Actually, you only need programs to implement construct and insert. For each implementation, estimate the average time to insert \( n \) elements into an empty priority queue. Describe the implementation you use, and show results supporting your estimate.