420-B31

# Lab 3

# Exceptions; Efficiency

## Answers

Part B, question 1 - Algorithm Description

public static int doSomething( int array[], int n ) {

int k = 0;

for ( int i = 1; i < n; i++ )

if ( array[i] > array[k] )

k = i;

int temp = array[0];

array[0] = array[k];

array[k] = temp;

k = 1;

for ( int i = 2; i < n; i++ )

if ( array[i] > array[k] )

k = i;

return array[k];

}

1. Set k to 0.
2. Loop through the array until the end.
   1. At each element, you check to see if the current element is larger than the element at k
      1. If the current element is larger, then k becomes the index of the current element.
3. Set a temporary variable to be the first element in the array.
4. Set the first element in the array to be equal to the last index of the current element
5. Set the last index of the current element to be equal to the temporary variable. (Swapping the two elements in the array)
6. Set k to 1.
7. Loop through the array again skipping the first two elements
   1. At each element, check is the current element is larger than the element at k.
      1. If the current element is larger, then k becomes the index of the current element.
8. Returns k.

The method returns the index of the largest element in the array I believe.

Page B, question 2 – Big O Notation

O(n)