1. (10 points) Given the following algorithm:

**Algorithm UniqueElements(A)**

- **Input:** n-element array of numbers, A[1...n]
- **Output:** Returns “true” if all items are unique and “false” otherwise

1. `for ( i = 1 to n−1 )`
2. `for ( j = i+1 to n )`
4. `return true`

(a) Is there a difference in $T(n)$ (measured by number of operations executed) for best- and worst-case input? If your answer is “yes”, give examples of best- and worst-case inputs. If your answer is “no”, explain why there is no difference between best- and worst-case inputs.

Yes there is a difference in $T(n)$ between best- and worst-case. In a best case input, the first 2 items checked in line 3 could be equal, in which case the for loop would be executed once. In a worst case input, all the items in the array are unique.

(b) Give the line number(s) of the basic operation. Express the running time using $\Theta$ notation if there is no difference between best-case and worst-case running times, or give best-case and worst-case asymptotic running time using $O$ and $\Omega$ if there is a difference.

Line 2 is the basic operation.

The best case running time is $\Omega(n)$. The worst case running time is $O(n^2)$.