1. Assume that $f(n) = O(n^x)$, $g(n) = O(n^y)$, $h(n) = O(n^z)$.

$f(n) = O(g(n)) = O(O(n^y))$

$f(n) \times h(n) = O(O(n^y)) \times O(n^z) = O(O(n^y) \times O(n^z))$

$O(g(n) \times h(n)) = O(O(n^y) \times O(n^z))$

$\therefore f(n) \times h(n) = O(g(n)) \times h(n))$

2. Assume that the claim is false. : n is $O(\frac{1}{n})$

If n is $O(1/n)$ then by definition, there are constant $c>0$ and $n_0 \geq 1$ such that

$n \leq c \cdot \frac{1}{n}$ for all $n \geq n_0$.

Simplify the inequality : multiplies both sides by n.

$n^2 \leq C$ for all $n \geq n_0$.

The inequality is valid only for values of n^2 that are at most c, so the inequality cannot be true for all values n larger than some constant n_0

Consequently, n is not $O(1/n)$.

4. i. a. Because the range of i is from 1 to $n-1$ and the range of j is from 1 to $i-1$

b. If all values stored in A are different then the algorithm must return true.

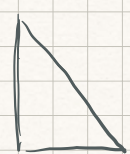
For all values stored in A are different, A[j] = A[i] will always be False and it will not return False.

For A that has two values are the same, A[j] = A[i] will be true and the algorithm will return False.

ii The worst case is all values are different or the last two values are the same then it need to go through all values.

iii Primitive operations : ←, =, return

	←	=	return
$i=1, j=0$	1	1	
$i=2, j=0$	1	1	
$i=2, j=1$	1	1	
---	---		
$i=n, j=n-1$	1	1	1
total	$\frac{n(n-1)}{2}$	$\frac{n(n-1)}{2}$	1



$t_n = n(n-1) + 1$ is $O(n^2)$

5.

n	Linear Search	n	Quadratic Search	n	Factorial Search
5	89 ns	5	231 ns.	7	3816100 ns
10	220 ns	10	694 ns.	8	19007600 ns
100	547 ns	100	15054 ns	9	89166700 ns
1000	7868 ns	1000	157858 ns	10	810013300 ns
10000	11698 ns	10000	13874551 ns	11	9280913400 ns
100000	49190 ns.			12	109253433900 ns.