@author Rufina Talalaeva

Bonus Question: If we were only interested in top K teams, we could easily modify selection sort to get top K. Just going in the first loop from 0 to K-1 and in the compare statement searching the max value into the beginning, because we do not need to swap other elements, except K bigger ones.

isContainedArray(A, B)	Cost	Times
1. for i ← 0 to n-1 do	c_1	n+1
2. contained ← FALSE	c_2	n
3. for j ← 0 to n-1 do	c_3	n(n+1)
4. if A[i] = B[j] then contained ← TRUE	c_4	n^2
5. if not contained then return FALSE	c_5	n
6. return TRUE	c_6	1

a. The running time of the algorithm is the sum of running times for each statement executed. The running time of this algorithm on an input n values:

$$T(n) = c_1(n+1) + c_2 n + c_3 n(n+1) + c_4 n^2 + c_5 n + c_6$$
$$T(n) = c_1 n + c_1 + c_2 n + c_3 n^2 + c_3 n + c_4 n^2 + c_5 n + c_6$$
$$T(n) = (c_3 + c_4)n^2 + (c_1 + c_2 + c_3 + c_5)n + (c_1 + c_6)$$

We can express this worst-case running time as $an^2 + bn + c$ for constants a, b and c that again depend on the statement costs c_i. Thus it is a quadratic function of n.

Definition from Cormen's section 3.1:

$O(g(n)) = \{f(n) :$ there exist positive constants c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0\}$.

Our function T(n) can be expressed as $f(n) = an^2 + bn + d$ & $g(n) = cn^2$.

$$0 \le an^2 + bn + d \le (a + |b| + |d|)n^2$$
$$0 \le a + \frac{b}{n} + \frac{d}{n^2} \le (a + |b| + |d|) \; for\ all\ n \ge 1$$

According to the definition of big-oh: $f(n) \in O(n^2)$.

b. $A = \langle 1,1,1,\dots,1\rangle$

If the algorithm is $\Omega(n^2)$, it means that T(n) should be at least n^2. This will happen iff two loops will execute completely. To happen this case we should prevent the return false case. Thus, in the array B should be at least one element that equals to 1. So, all arrays B with length n and containing at least one element 1 will produce the $\Omega(n^2)$ runtime.

$$B = \langle 1, any, any, \dots, any\rangle$$
$$B = \langle any, 1, any, \dots, any\rangle$$
$$B = \langle 1,1, any, \dots, any\rangle$$
$$B = \langle 1,1,1,\dots,1\rangle$$

c. Theorem from Cormen's section 3.1:

Theorem 3.1

For any two functions $f(n)$ and $g(n)$, we have $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$. ■

According to a., c. and theorem 3.1: $f(n) = O(n^2) and\ f(n) = \Omega(n^2)$ <=> $f(n) = \theta(n^2)$

Yes, the worst-case runtime of the algorithm is $\theta(n^2)$.