Quiz 2 **Algorithm Section B – FALL 2016**

Q 1: Find time complexity of following recurrences using Master method. Mention which case applies and show working. (5)

1) $T(n) = 3T(n/2) + n$
2) $T(n) = 0.5T(n/2) + 1/n$
3) $T(n) = 16T(n/4) + n!$
4) $T(n) = 3T(n/3) + \sqrt{n}$
5) $T(n) = 4T(n/2) + n^2$

Q 2: Write upper bound, Big-Oh, O() of following time complexities. (2)

$T(n) = 10n^3 + 4n^2 + 16n$

$T(n) = n\log n + n^2 + 100$

Q 3: Write an algorithm (using C/C++/Java or pseudocode as shown in class) to **find 2nd largest element in an unsorted array**. Give time complexity of your algorithm. Don't use any built in functions. To swap 2 numbers you may use swap(i, j) method. Method should return the 2nd largest element. (5)

```
int Find2ndMax(Arr) {}
```