4. a) Monte Carlo: Choose random k s.t. $0 \leq k < n$
Pick A[k] as dominant element.

Picking a random element is O(1). Since the element is dominant in the array, there is a $> \frac{1}{2}$ probability to pick the right one.

b) las-vegas (A, n)
A: array, n: size of A
k ← random(0, n) // $0 \leq k < n$
elem ← A[k]
count ← 0
for i ← 0 to n-1 do { if (count > n/2) return elem
if elem == A[i] {
count++
} else {
if count != 0
count --
else {
elem ← A[i]
count ← 1
}
}
}
return elem

If A[k] is indeed dominant, the counter will always remain positive and we'll return A[k]. If it is not, the counter will eventually reach 0 as the # dominant elements $\geq \lfloor n/2 \rfloor + 1$, and the variable elem would become the dominant element. If counter is ever $> n/2$, we return the element.
Since we loop through the array once, at max. run-time is O(n).