My numbers:
18040180

336362202432400

3620415789

Q2. 3620419578

Q3. pictures
Q4. pictures
Q5. pictures

Q6.
In Asymptotic Analysis, we evaluate the performance of an algorithm in terms of input size, we don't measure the actual running time. We calculate how the time or space taken by an algorithm increases with the input size. When we use asymptotic notation, we are talking about worst-case running time unless mentioned otherwise. Big Oh notation is used to describe the asymptotic upper bound of an algorithm, it's the measure of the longest amount of time it could possibly take for the algorithm to complete. Big Omega notation is used to describe asymptotic lower bound of an algorithm, it's the smallest amount of time it could possibly take for the algorithm to complete. Big Theta notation is used to define the running time of an algorithm, it gives us a tight bound because the running time is within a constant factor of above and below. The bounds Big O and Big Omega can be tight or loose, but we prefer to make them tight as possible. If we have tight bounds where Big O and Big Omega are the same then we have the growth rate or Big Theta. In web request companies like facebook would use big O notation to ensure that everyone is getting a decent quality of service all of the time. For algorithms where big theta is known like binary search we would use big Theta notation.

Q7.
I am using Asymptotic notation.

Let nums.length = M.

Assume size() = big-theta(1)

Line 1 is big-theta(1) and is executed big-theta(1), this implies T1 = big-theta(1).

Line 2 is big-theta(1) and is executed big-theta(1), this implies T2 = big-theta(1).

Line 3 is big-theta(1) and is executed big-theta(1), this implies T3 = big-theta(1).

Line 4 is big-theta(M) and is executed big-theta(1), this implies T4 = big-theta(M).

Lines 5 is big-theta(logk) and is executed big-theta(M), this implies T5 = big-theta(Mlogk)

Lines 6 is big-theta(1) and is executed big-theta(M), this implies T6 = big-theta(M)

Lines 7 is big-theta(logk) and is executed big-theta(M), this implies T7 = big-theta(Mlogk)

Lines 8 is big-theta(1) and is executed big-theta(1), this implies T8 = big-theta(M)

The total running time for T(N) = T1 + T2 + T3 + T4 + T5 + T7 + T8, we keep only the higher order terms so the asymptotic running time, T(N) = big-theta(Mlogk) .

The amount of space used is k+1