Recurrence Relation

Assume $T(n)$ is a runtime of an algorithm written as a recurrence relation. Find $\theta(T(n))$ by guessing the number of operations needed to complete each algorithm. Assume that $T(n) = c, \quad n \leq d$ where c, d are constants.

1. $T(n) = T(n-5) + c * n^2$
2. $T(n) = 2T(\frac{n}{3}) + h(n), \quad h(n) = cn^3$
3. $T(n) = 5T(\frac{n}{5}) + h(n), \quad h(n) = cn$
4. $T(n) = 3T(\frac{1}{6}) + c$
5. $T(n) = T(\frac{n}{5}) + T(\frac{n}{7}) + c$ Find upper and lowerbound separatly.

Finding Peak Element

You are given an array A of n distinct elements, and there is exactly one i in this array s.t.

$A_j > A_{j+1}, \quad j < i$

$A_j < A_{j+1}, \quad j >= i$

For example peak of $A = [50, 32, 17, 5, 9, 11, 90, 1671, 8888]$ is number 5 in index 4. Proposed a divide and conquer algorithm to find the index of i with the properties stated above, analyze your algorithm.