1. n^3-$3n^2$-n+1 < n^3 for all n>=1
   n^3 < 2 (n^3-$3n^2$-n+1) for all n>=4
   so, n^3-$3n^2$-n+1=Θ(n^3)

2. Find-the-gap({a_1,…a_n})

```
        if (a_1 != 1), return 1
        else if a_n =n, return n+1
        else
                start = 1, end = n
                while (start < end)
                        i = ( start + end ) / 2
                        if a_i=i, start= i+1
                        else, end =i
                return a_{start-1} + 1
```

3. a) Run BFS to find all the reachable vertices from the given vertex v

   if the vertex queue is not empty, then v is not a mother vertex.

   b) If G is a directed acyclic graph (DAG), then the only possible candidates for the mother vertex are the vertices with no incoming edges.

   Find all such vertices in O(|V|+|E|) time. If there are more than 1, then G has no mother vertex.

   If there is only one such vertex, then follow (a) to check whether this vertex is the mother vertex.

4. Our goal is to find the two longest paths to a leaf from the any node r in the tree. The diameter of the tree is the sum of these two lengths (starting from one leaf u, reaching up until r, then again go down until another leaf v). Start from each leaf node, go up one level at a time to the parent node and keep track of its height/ maximum distance from the leaves.

   T_r = Tree rooted at vertex r
   D (T_r) = diameter of the tree rooted at r
   L (r) = Maximum length path to a leaf node from r

   L(r) = 0 , if r is a leaf node
   = max (L(v)) + 1, where v $\in$ child(r)
   D (T_r) = $\max_{u, v}$ (L (u) + L (v) + 2) where u,v $\in$ child(r)

   We have to compute D(T_r) for all subtrees T_r in the graph. We need to examine each vertex once. Therefore, the running time will be O(n).