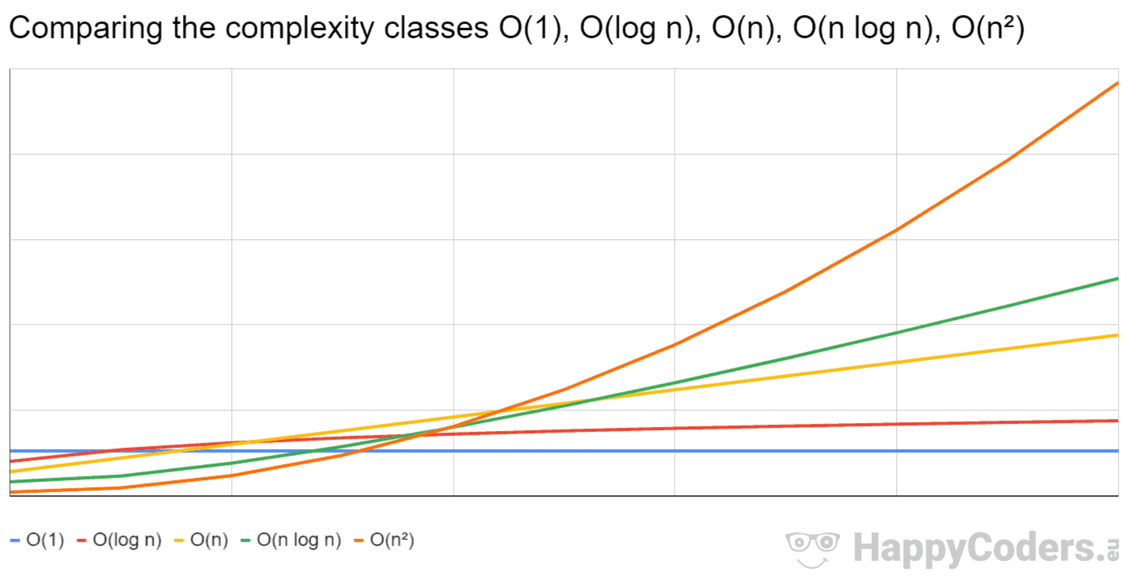
Quiz 6 Aditya Shankar – 5454360

Q1. The main reason to use multilevel graph partitioning is to reduce the time complexity of the partitioning algorithm. The idea of using recursively smaller problems to achieve faster computation can be found in various other algorithms: e.g., Merge Sort and multigrid.

Recursion reduces the time because it introduces a "log" factor that grows at a smaller rate compared to polynomial time complexities for large sizes N. The figure below illustrates this. Notice that O(NlogN) has a smaller growth rate compared to higher order polynomials like O (n^2).



Non-multilevel graph partitioning methods like BFS and Inertial partitioning have a polynomial time complexity, so for large graph sizes they will not perform well. Through multilevel partitioning, the partitioning problem can be broken down to a simpler problem recursively and then solved. After solving the simpler problem, the results can be transferred back to the original graph.

Q2. The main steps in the Barnes- Hut algorithm are as follows:

1. Building of a QuadTree (or Adaptive QuadTree). To represent the nodes/particles in the system. Time complexity is O(bN), where b is the number of bits to represent the node's coordinates and N is the number of nodes. This translates to O(NlogN) for uniform distribution (depth of the tree)
2. Computing the total mass and centre of mass for each square in the QuadTree. This involves a post-order traversal of the tree and therefore has a time complexity of O(NlogN) or O(bN)
3. Force calculation for each node is then computed by considering all other nodes. If the distance of the node to the corresponding square's centre of mass is not large enough, the total force is computed by summing the force on the node due to all the other children within that square. Else, the centre-of-mass of the other points can be used to estimate the net force. This time complexity is also of order O(NlogN) or O(bN)

Comments on the time complexity reduction.

The time complexity is reduced from O (n^2) to O(NlogN). However, this depends on the value given for the threshold that determines how far apart the centre-of-mass of a set of nodes needs to be from the particle under consideration, i.e., theta = D/r, where D = Distance between CM and node, and r = box dimension. If this value is zero, then pairwise computations must be done for all particles and so the complexity is of O (n^2).

Q3. The previous answer explains how Barnes-Hut achieves a time complexity reduction from O (n^2) to O(NlogN). This involved three steps:

1) Tree construction

2) CM computation, and

3) Force computation.

Each of these steps has a time complexity of O(NlogN), so if performance must be further improved by parallelism, then all three steps have to be parallelized.

However, in a parallel algorithm, each processor requires the entire QuadTree, and constructing this tree has complexity of O(NlogN), which would make moot any gains made in optimizing just steps 2 & 3. This is the main challenge in parallelizing the Barnes-Hut algorithm.

The problem of having to optimize all the steps was not an issue with multigrid, making the parallelization of Barnes-Hut an even more challenging problem