<https://www.youtube.com/watch?v=V6mKVRU1evU>

you can also check this video for better explaination

BIG-O NOTATION

You have, actually, three kinds of algorithms, and not much more:  
  
1. O(n): Linear (all functions of the like: f(x)= kX + C, and the like)  
2. O(N \* log N): logarithmic grow (anything that entails trees usually)  
3. O(n^x): exponential. (Or a polinomial, like: kn^x + ... nm + i; too costly, avoid this at all cost)

A few concrete examples. Just say I'm in front of a class of students and one of them has my pen. Here are some ways to find the pen and what the O order is.  
  
**O(n^2 ):** I question a student and ask them, "Does Jeff have the pen? No? Does Bob have the pen?" And so on, naming each student. If I don't get the answer from the first student, I move on to the next one. In the worst case I need to ask *n^2* questions - questioning each student about each other student.  
  
**O(n):** I ask each student if they have the pen. If not, I move on to the next one. In the worst case I need to ask *n*questions.  
  
**O(log n):** *I divide the class in two, then ask: "Is it on the left side, or the right side of the classroom?" Then I take that group and divide it into two and ask again, and so on. In the worst case I need to ask log n questions.*  
*I might need to do the O(n^2 ) search if only one student knows on which student the pen is hidden. I'd use the O(n) if one student had the pen and only they knew it. I'd use the O(log n) search if all the students knew, but would only tell me if I guessed the right name.*  
  
**O(1):** Which one of you have my pen?