**Big O**

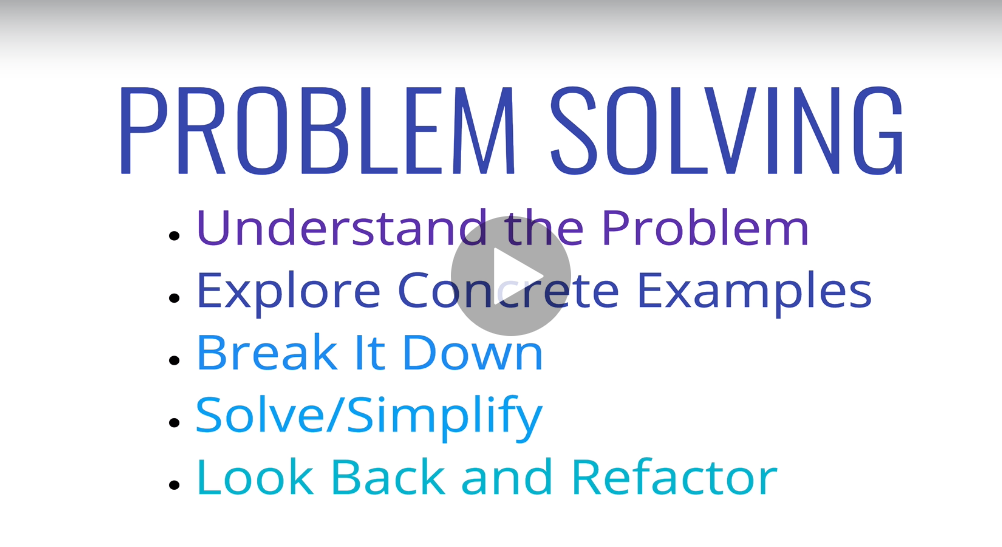
Big O: We say an algorithm is O(f(n)) if number of simple operation the computer has to do is eventually less than f(n) if n increases

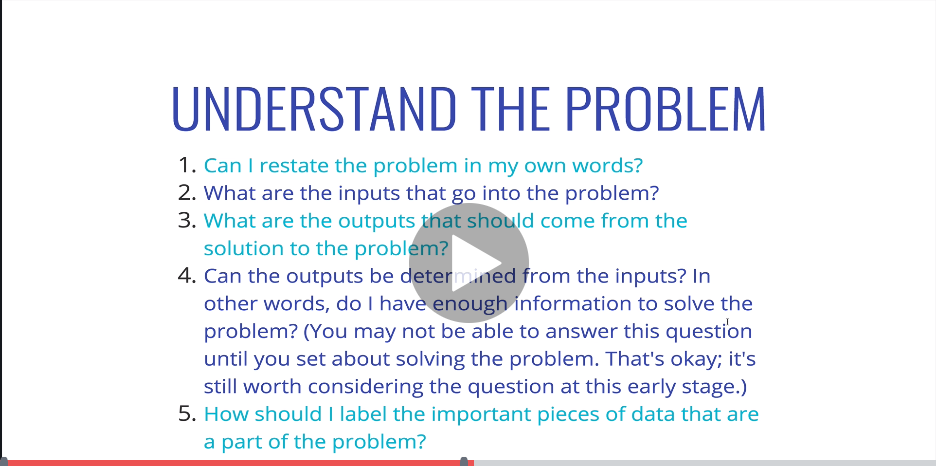
1.f(n) could ne linear (f(n) = n)

2.f(n) could be quadratic (f(n) = n^2)

3.f(n) could be constant (f(n) = 1)and so on

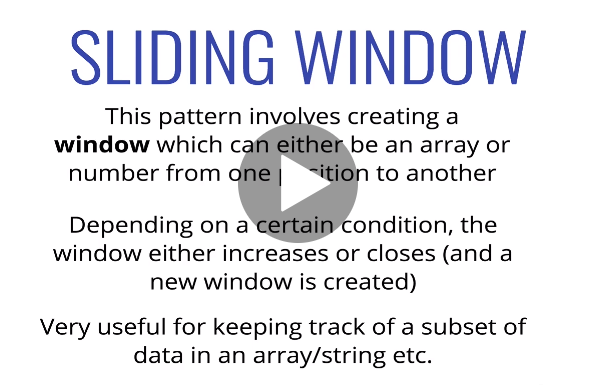
**Problem Solving**

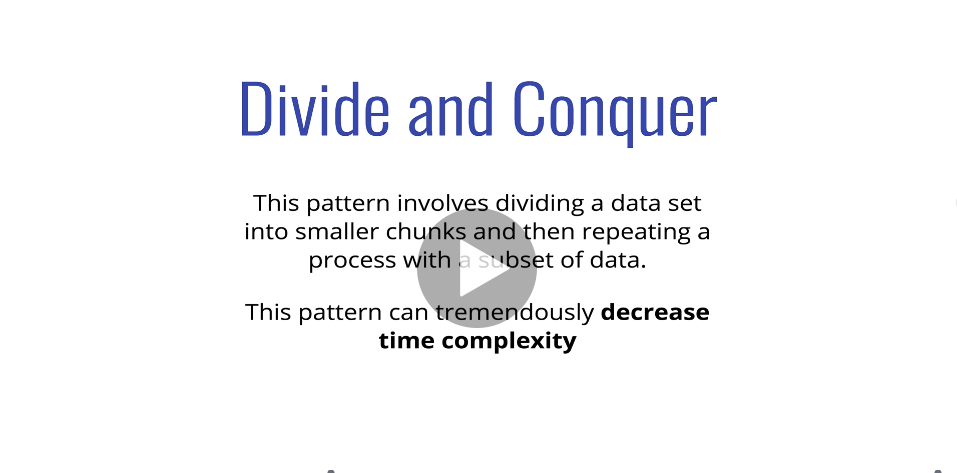




Problem Solving patterns

1. Counter problems – Use Javascript objects to avoid O(n\*2)
2. Multiple pointers problem – Keep multiple pointers to avoid repeating of loop
3. Sliding window pattern – Useful for solving problems which includes subset
4. Dynamic programming – Sometime cache data if required for future use





Recurssion: function calling itself

