Heaps:

* Insert
  + Insert at next position which is in the lowest and furthest left location
  + Upheap: Compare with parent and swap if necessary
* Remove
  + Replace root with tail
  + Downheap: Compare with children. This is the one that finds the deepest child
* MinHeap: top is the smallest
* MaxHeap: top is the largest

BST:

* Worst Case is O(n)

Time Complexities

* lim(f/g) = 0 means that f is Big O of g (Worst Case)
* lim(f/g) = c means that f is Big Theta of g (Average/Amortized Case)
* lim(f/g) = inf means that f is Big Omega of g (Best Case)

NP-Complete:

* Problems that are really computationally expensive, usually at exponential time (Travelling Salesman, Knapsack, Hamiltonian Path)
* You can transform NP Complete problems into any other NP complete problem in polynomial time