现在对算法的认识：

1.我基本认同P != NP, NPC = intersect of NP-Hard and NP

这一类问题没法解，所以只能用近似算法和随机算法来解决。而大部分人工智能，机器学习和数据挖掘的问题都属于这一类问题（？）。

2.算法的核心其实是：问题的分析与策略的设计

设计的几大类：

- Greedy

e.g., Dijkstra, Kruskal, Prim, Huffman code,

- Induction: D&C, DP

- Backtracking / Branch & Bound

3.数据结构

- 基本数据结构(array, list, tree)

- 其他所有的更高级的DS都是由基本的演变过来的，它们只不过是在基本DS的基础上增加了一系列的规则而已

- 也就是说，实现的时候还是用这几个来实现的

- 高级的DS能够更好地契合问题的模型

- 图作为一种特殊的DS，能够对非常多的问题进行抽象，同时又有自己独有的十分巧妙的算法

作用

- 用来储存数据

- 算法实现过程中的辅助作用

4.剩下的就是数学问题和特殊的问题了，不纠结

- Linear Programming (in Mathematical Programming)

- Matrix Multiplication

- 字符串操作

- 大数操作

- etc