

整理思路：

- 1、又看到了 NP 问题，找了一些资料，然后整理理解一下，便于以后的翻阅。
- 2、先看定义：直接上图，看不懂下面的图，真的看不懂。

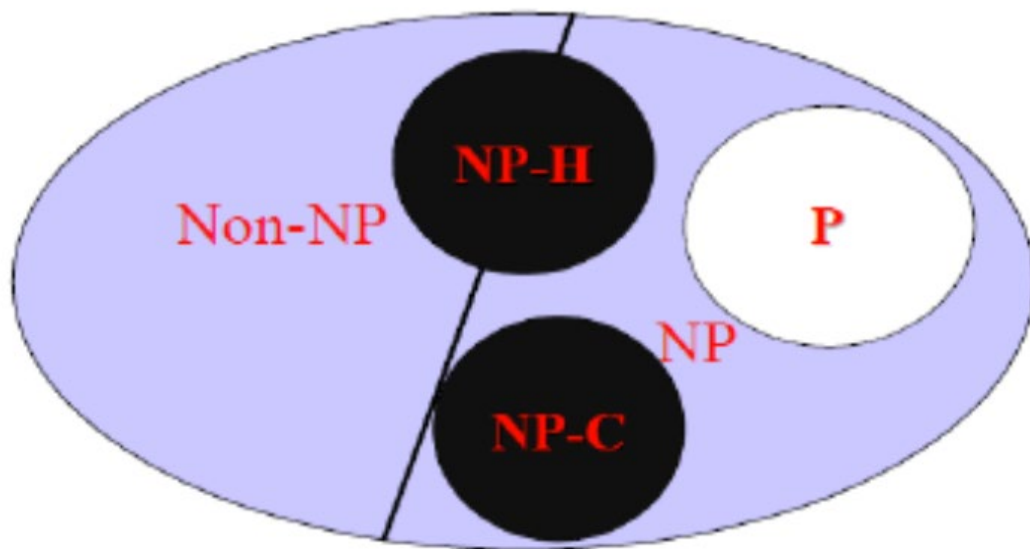


图 1 各类问题的集合论关系

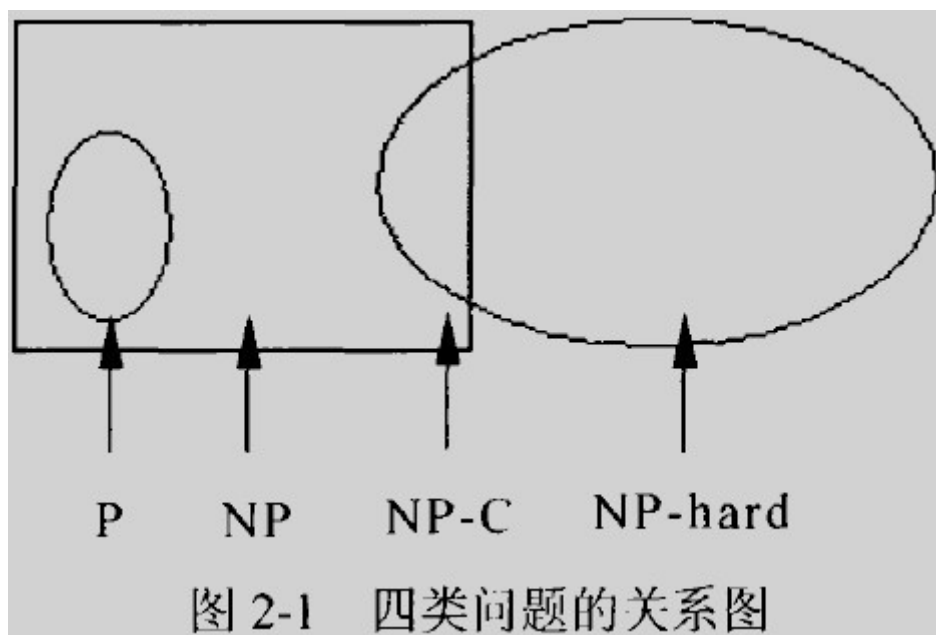
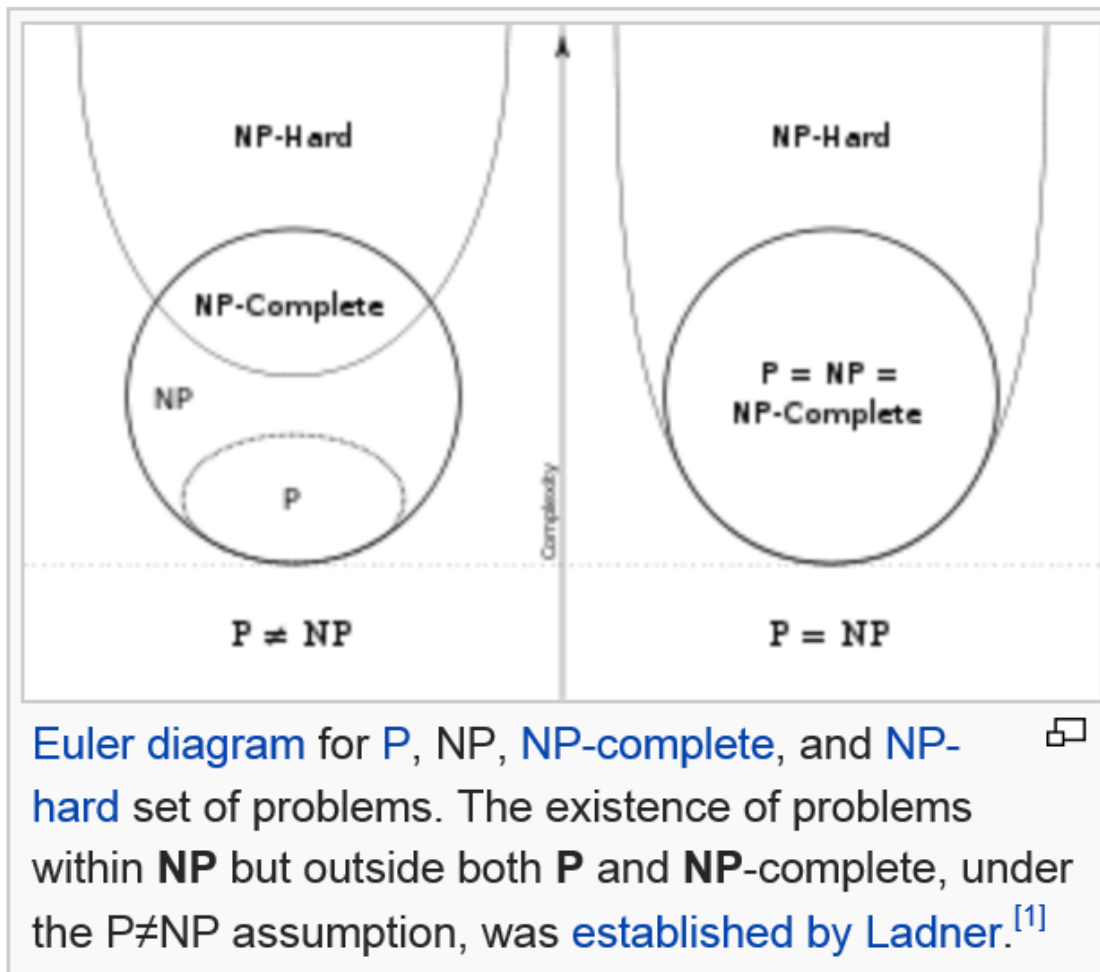


图 2-1 四类问题的关系图



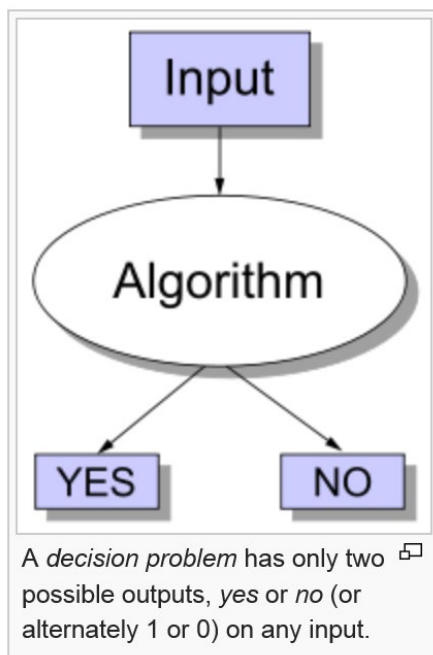
3、上图直接把四类问题给出来了，但是真的不理解。因此要明白，问题是怎么提出来的，或者说怎么引申出来的。看例子，截图：

在一个周六的晚上，你参加了一个盛大的晚会。由于感到局促不安，你想知道这一大厅中是否有你已经认识的人。你的主人向你提议说，你一定认识那位正在甜点盘附近角落的女士罗丝。不费一秒钟，你就能向那里扫视，并且发现你的主人是正确的。然而，如果没有这样的暗示，你就必须环顾整个大厅，一个个地审视每一个人，看是否有你认识的人。

这里牵涉出一个概念，就是这个问题是让你回答：有还是没有，或者对还是不对（“针对这里面有没有你认识的人”）。这个问题，就是“判定问题”。看截图：

定义2.1 如果一个问题的每一个实例只有“是”或“否”两种答案，则称这个问题为**判定问题** (Decision / recognition / feasibility problem). 称有肯定答案的实例为“**是**”实例 (yes-instance). 称答案为“否”的实例为“**否**”实例或非“是”实例 (no-instance).

继续看英文的截图：



4、那么 NP 和这个“判定问题”有毛关系呢？看截图说明：

评价一个算法的依据是该算法在**最坏实例下**的计算时间与实例输入规模的关系： $C(I) \leq \alpha g(d(I))$ or $C(I) = O(g(d(I)))$
存在多项式函数 $g(x)$ 满足上式时，算法为多项式算法

存在多项式算法的问题集合：多项式问题类 (P)

比多项式问题类可能更广泛的一个问题类是非确定多项式 (**Nondeterministic Polynomial**，简记 NP) 问题类

NP 类是通过判定问题引入的。

NP 问题是通过判定问题引入的。怎么引入的，我们分析一下。

分析不来啊，没有办法，从另外一个角度去引入吧。

5、就从字面意义上理解，确定和非确定。看截图：

什么是NP(非确定性问题)呢？有些计算问题是确定性的，比如加减乘除之类，你只要按照公式推导，按部就班一步步来，就可以得到结果。但是，有些问题是无法按部就班直接地计算出来。比如，找大质数的问题。有没有一个公式，你一套公式，就可以一步步推算出来，下一个质数应该是多少呢？这样的公式是没有的。再比如，大的合数分解质因数的问题，有没有一个公式，把合数代进去，就直接可以算出，它的因子各自是多少？也没有这样的公式。

简单说，有些问题，你可以通过确定的公式和某种算法，得到结果。相反的，有些问题，你是没有确定的方法得到答案，你只有通过枚举法，但是这样的话，可能会无穷的，因此，计算量是非常大的。那只能猜测了，运气好的话，一次就中，运气不好的话，...。因此这里就有了两种说法：一个 P 类问题，是我可以在“多项式时间”里面，通过每种方法得到解；一个 NP 类问题，是我可以在“多项式时间”里面，去验证我给出来的一个解，是不是我要的解，或者说是对的还是错的（回到了“判定问题”这个概念）。

那就要举例说明，先看 NP 问题，截图上：

NP问题是指可以在多项式的时间里验证一个解的问题。NP问题的另一个定义是，可以在多项式的时间里猜出一个解的问题。比方说，我RP很好（^.^），在程序中需要枚举时，我可以一猜一个准。现在某人拿到了一个求最短路径的问题，问从起点到终点是否有一条小于100个单位长度的路线。它根据数据画好了图，但怎么也算不出来，于是来问我：你看怎么选条路走得最少？我说，我RP很好，肯定能随便给你指条很短的路出来。然后我就胡乱画了几条线，说就这条吧。那人按我指的这条把权值加起来一看，嘿，神了，路径长度98，比100小。于是答案出来了，存在比100小的路径。别人会问他这题怎么做出来的，他就可以说，因为我找到了一个比100小的解。在这个题中，找一个解很困难，但验证一个解很容易。验证一个解只需要 $O(n)$ 的时间复杂度，也就是说我可以花 $O(n)$ 的时间把我猜的路径的长度加出来。那么，只要我RP好，猜得准，我一定能在多项式的时间里解决这个问题。我猜到的方案总是最优的，不满足题意的方案也不会来骗我去选它。这就是NP问题。