计算机科学课程

经过一段时间的顾问工作和聊天沟通，比较深入的了解到国内计算机教育的现状和大家对于学习的困惑，我觉得是可以“系统授课”的时候了。只是这次的授课，恐怕和我原来设想的方式有很大不同。

每当需要“系统”的准备任何事情，我都会犯严重的拖延症。一方面是我其实不想对别人负责，而且“系统”这个词的含义，对于我来说也很模糊。心里有了“教好这门课”，“系统化”…… 这些目标，反而很难真正开始准备课程。我要教他们什么内容呢？从哪里开始？要是我没理解他们，他们听不懂怪我呢？做幻灯片好麻烦啊，精益求精…… 一系列的紧张。

最后经过思考，我发现自己最有效的工作方式，其实不是系统而周密的准备，而是即兴的发挥，边做边想。不管是参加我的顾问服务，还是加入我的聊天室的人，都发现学到了很多。他们是从对话中去领悟，而不是传统的课堂。我不是作为一个“真理传播者”，而只是一个“启发者”。我甚至可以有些时候头脑不清楚，概念也稀里糊涂，而且很多的“新概念”我其实没仔细看过，我甚至可以完全是错的。

但通过对话和提问，我们产生了头脑里从来没有过的知识，而且深刻理解了它的来龙去脉。这种认识可能超越文档或者书籍。

这就是对话的力量，这让我再一次想起“Little 系列”书籍。为什么《The Little Schemer》是对话的形式，为什么它用那么短的篇幅，可以教会我比普通书籍多很多的东西？我一直很好奇这个现象。

有一次我问 Friedman，你的书怎么都是这种对话风格啊？这叫什么 style？他笑了：“这叫 little style！” 后来我发现，这种对话式的教学方式几千年前就出现了，叫做「苏格拉底方法」。

> 苏格拉底承认他自己本来没有知识，而他又要教授别人知识。这个矛盾，他是这样解决的：这些知识并不是由他灌输给人的，而是人们原来已经具有的；人们已在心上怀了“胎”，不过自己还不知道，苏格拉底像一个“助产婆”，帮助别人产生知识。

孔夫子似乎也用了不少对话式的教学。不过孔子的方式是学生提问，老师只是回答，像顾问似的；而苏格拉底的方式是老师提问，学生回答。方向是相反的。

我个人觉得苏格拉底的方式要好些，因为苏格拉底方法中的老师并不是“布道者”或者“圣人”的角色，而只是一个“启发者”。对于某些话题，老师可以本来没有知识，但是老师很会提问，通过追问和对话，最后大家都得到了新的知识。另外，没有基础的学生经常不知道该问什么问题，所以老师提问可以启动学生思考。

没想到两千多年前，有人认识到比我们深刻很多的教学原理，而且它在我的实践中被印证了。每一次有意义的对话，都是一场头脑的 SPA，让人身心愉悦。当初我是作为学生，现在我是作为“老师”。

我发现“老师”和“学生”的界限越来越模糊。到底是谁教会了谁，也许其实并没有谁教会了谁？是的，作为一个“老师”，很多时候我并没有知识，可是通过对话式的教学，却产生了知识。

Friedman 的情况也类似。他问你的有些问题，可能他自己都还没想清楚，但他经常问很好的问题。跟他对话的时候，他不总是对的，有些时候甚至稀里糊涂的。可是他会指引你朝着破除谜团的方向前进，最后他和你一起把问题想得清清楚楚。

其实有好几本新的 Little 书，都是 Friedman 开头不熟悉的主题：《The Reasoned Schemer》，《The Little Prover》，《The Little Typer》…… 可是通过与合作者讨论，从他们身上学习，找该领域最高明的专家咨询，他把这个领域给嚼烂消化掉，然后把精华的营养都写进了书里。

鉴于如此成功而快乐的试验，我决定不再试图准备一堂“系统”的课程，而是像爵士乐手一样即兴发挥。我授课的内容，授课的方式，都可能即兴改变。