

What have we learned on the course Algorithm Design?

dbu@ict.ac.cn

2019

How to solve problems?

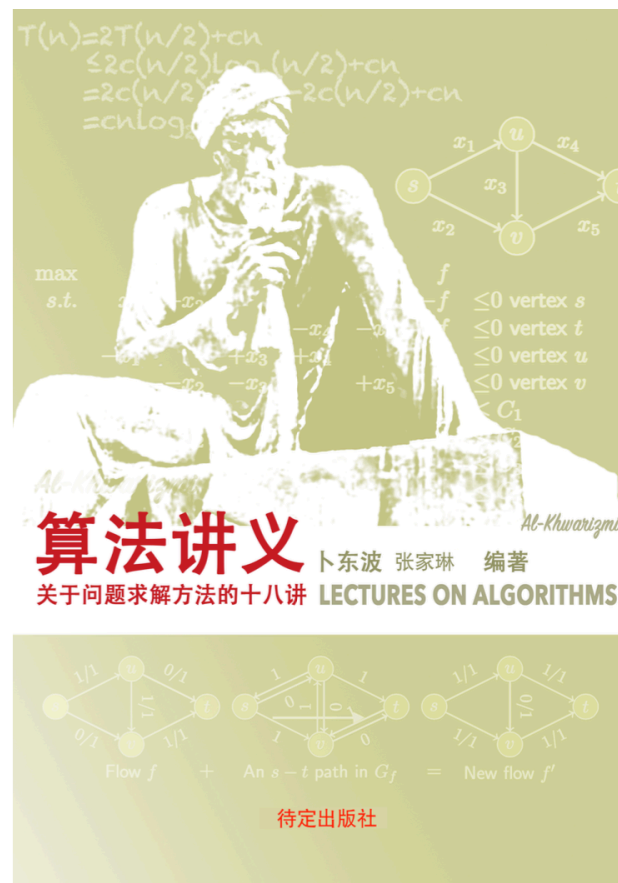
- **观察一、问题的可分解性：** 问题的最简单实例是什么？复杂的实例能否分解成简单的实例？
- **观察二、可行解的形式以及解之间的变换关系：** 问题的可行解的形式是什么？可行解的总数有多少？可行解能否一步一步地逐步构建出来？我们能否对一个可行解施加小幅扰动，将之变换成另一个可行解？
- **观察三、类似的问题：** 和给定问题类似的问题有哪些？解决类似问题的算法能否直接应用于解决当前的问题？如果不能，那又是什么因素造成了妨碍？能否想办法消除这些妨碍因素？
- **观察四、现有算法的不足：** 现有的求解算法有哪些不足？比如：是否存在冗余计算？如果存在的话，能否想办法去除冗余计算？

Several lessons

1. 基于对问题结构观察的算法设计;
2. 从最简单的例子做起;
3. 试图把大的问题分解成小问题;
4. 试图从粗糙解开始逐步改进;
5. 试图枚举所有的解, 但是 “being smart”;
6. 难以优化的函数, 用下界或上界函数替代, e.g., EM, Lagrangian, log barrier
7. 复杂操作的潜力一定要挖尽, e.g., Dinic's algorithm
8. 求同时满足多个条件的解, 分步满足, 并利用对称性, e.g., 3 strategies to solve LP, Push relabel;
9. 想想对偶, e.g., primal and dual;
10. Random sampling, e.g., select a pivot;
11. Scaling + rounding, e.g., Scaling FF, closest-pair;
12. Best => sufficiently good, e.g., select a pivot, scaling FF, push relabel;

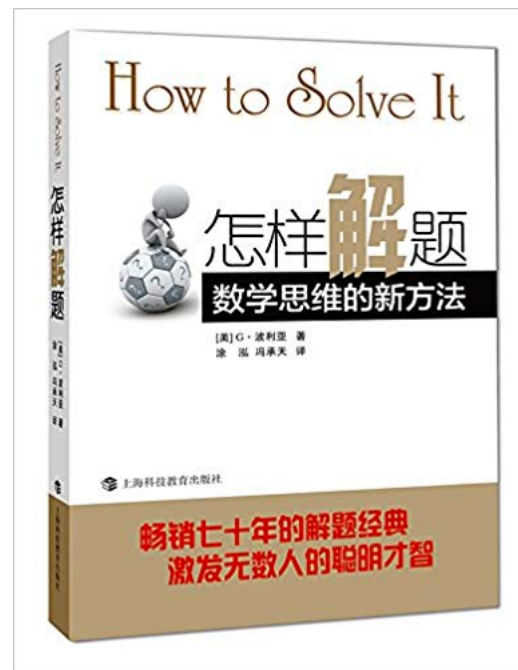
Acknowledgements

Thanks the scribes



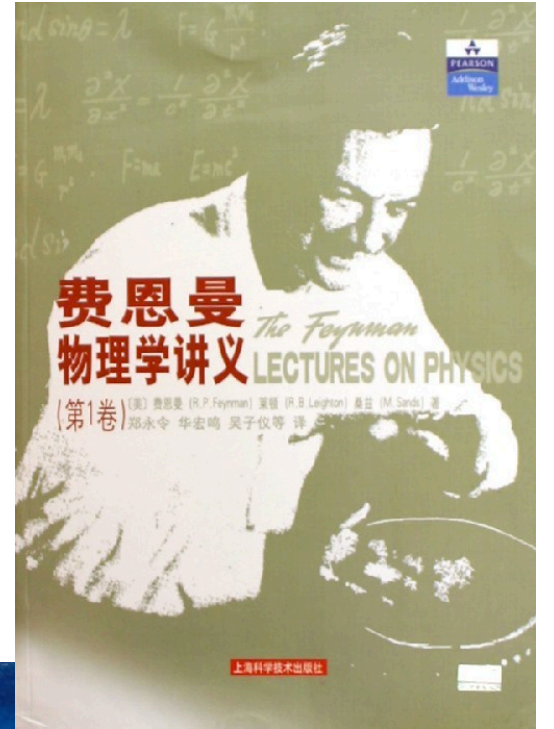
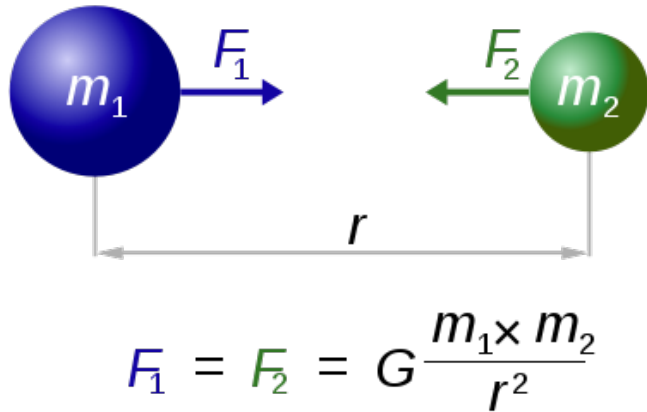
用这种写法的一次尝试 — 由中国科学院大学的同学们依据课堂录音整理成文；我们在此对乔扬、申世伟、邵益文、黄斌、闫泽军、乔晶、袁伟超、李飞、孔鲁鹏、吴步娇、张敬玮、罗纯龙、曹晓然、梁志鹏、江涛等同学致以谢忱。

Polya的《怎样解题》



当面对一个问题时，从何处入手进行求解值得深入思考。数学家 G. Polya 如此描述他学生时代的困惑：“是的，这个解答看起来是可行的，它似乎是正确的，但怎样才能想到这样一个解答呢？”。不仅要理解解答本身，而且要理解如何做出这个解答，并尽力向别人解释清楚思考过程—这些动机最终驱使 G. Polya 写出了关于数学思维的名著《怎样解题》。

What is a good book?



What is a bad book?



1980年代初，杨振宁曾在韩国汉城作物理学演讲时说「有那么两种数学书：第一种你看了第一页就不想看了，第二种是你看了第一句话就不想看了」。当时引得物理学家们轰堂大笑。此话事出有因。1969年，杨振宁察觉物理上的规范场理论和数学上的纤维丛理论可能有关系，就把著名拓扑学家Steenrod着的「The Topology of Fibre Bundles(纤维丛的拓扑)」一书拿来读，结果是一无所获。原因是该书从头至尾都是定义、定理、推论式的纯粹抽象演绎，生动活泼的实际背景淹没在形式逻辑的海洋之中，使人摸不着头脑。

Thanks Loongson

- These software run smoothly on Loongson!
- Text processing:
 - TexShop, Latex,...
 - Word, Powerpoint, evince, ...
- Mathematical packages:
 - GLPK
 - Octave (MatLab)
- Image processing:
 - GIMP, gnuplot,
- Programming languages:
 - Python, perl, gcc, g++, ...
-



Thank you all!