

路径规划那些事

14计算机2 项闰治

2018年4月25日

目录

- 引
- 路径规划
- 大学感悟
- Q&A



路径规划是什么？

- 导航
- 输入一个上车点和一个下车点，返回一条路径。

拼车 VS. 快车

- 旅行商问题 (TSP) NPC问题 时间复杂度
- 有最优解

人工智能 算法 滴滴出行 AlphaGo

如何看待滴滴章文嵩称，滴滴的算法比 AlphaGo 复杂 100 倍？

事件快讯

【滴滴派送体验不好？滴滴副总裁：滴滴的算法比AlphaGo复杂很多倍】滴滴副总裁章文嵩：滴滴的算法要考虑43200步。围棋格子里面只有19层棋局，最多361步，而且都有确定性的解，但滴滴调度的最优解是在不停变化的，所以这个问题比Alpha Go面临的问题要复杂很多很多倍，需要非常强大的科技手段和计算能力。

[滴滴副总裁：围棋最多361步 滴滴算法复杂100倍](#)

滴滴目前实力



滴滴文嵩 : 大家提到了分区来处理的工程解法，如以2公里为半径画一个圆，面积约12平方公里，里面有非常多订单需求和可用司机需要计算相关性。工程上实际处理的复杂度已经很高，约1年前滴滴对外公布的路径规划量是200亿次一天，大部分是分单系统调用的，平摊到每一秒约23万次，高峰时一秒钟处理1百万次路径规划。

现实世界抽象

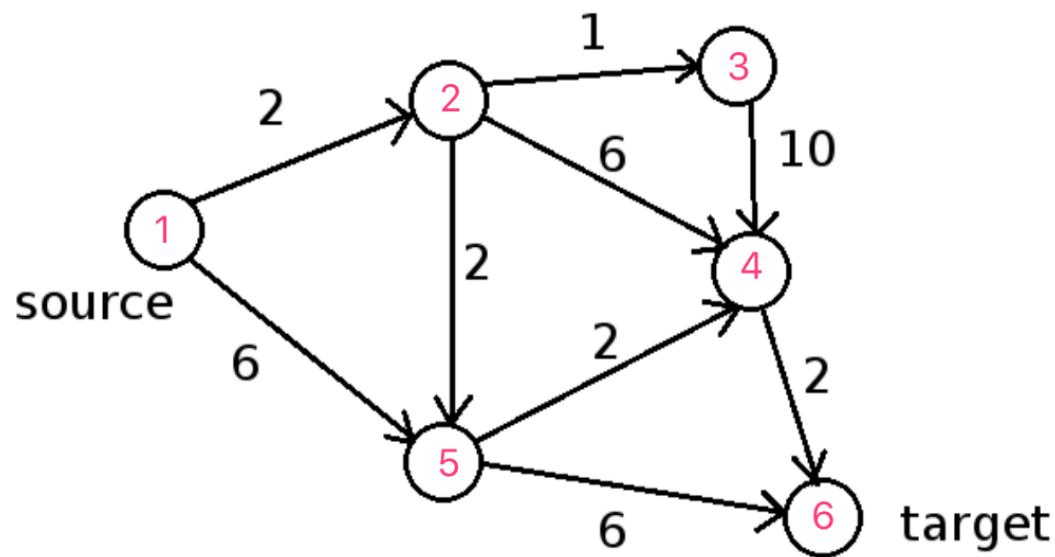


数据格式



序号	字段设置			取值范围	注释	
	含义	名称	类型			
1	组号码	VirConnID	Char(13)			
2	第一个点图幅号	MapID1	Char(8)			
3	第一个点号码	NodeID1	Char(13)			
4	第二个点图幅号	MapID2	Char(8)			
5	第二个点号码	NodeID2	Char(13)			
6	类型	Type	Char(1)	1	直梯	
				2	扶梯	
				3	阶梯	
				4	斜坡	
				5	索道	目前无数据
				6	复合类型	
				7	地下通道	
7	坡度	Grade	Char(1)	1	上坡	第一点到第二点方向 Type 为阶梯和斜坡时有效
				2	下坡	
8	通行方向	TravelDir	Char(1)	1	双方向	Type 为扶梯时有效;
				2	第一点到第二点方向	
				3	第二点到第一点方向	
9	复合类型属性	LoctType	Char(1)	1	建筑物	Type 为复合类型时有效;
				2	免费公园	
				3	收费公园	
				4	广场	
				5	街道	
10	复合类型是否存左暗码(仅楼梯)	Stairs	Char(1)	0	不存在	
				1	存在	

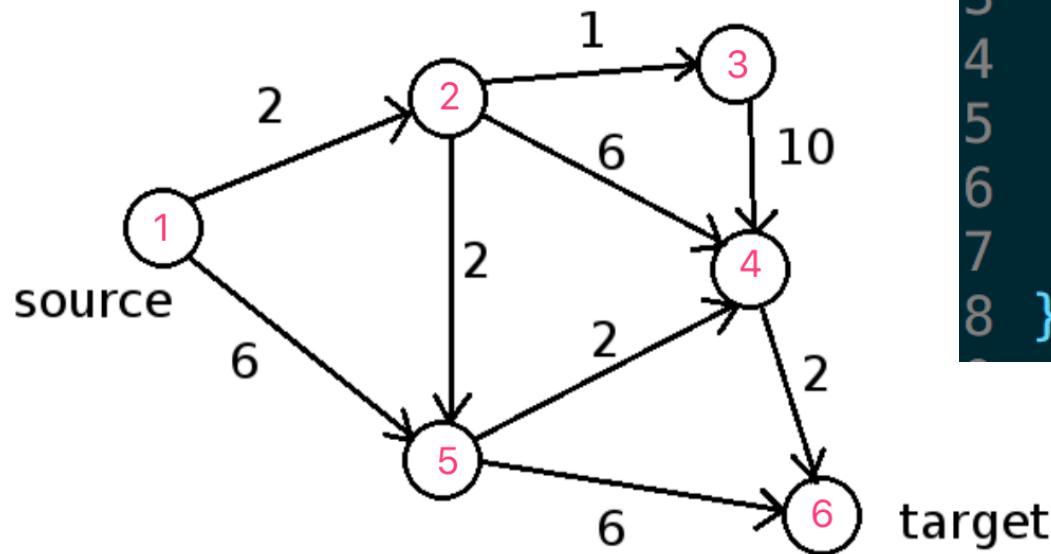
课本中的路径规划



图论

- 点
- 边
- 权值
- 方向

图的数据结构描述



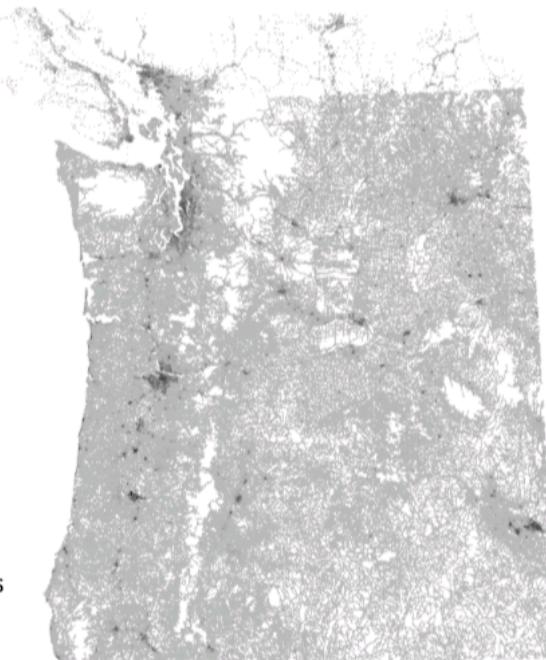
最短路问题

```
1 struct edge
2 {
3     int start_id;
4     int end_id;
5     int edge_id;
6     bool direction;
7     int weight;
8 }
```

找出从点 1 到 点 6 的
权值和最小的路径

现实

Example Graph



最短路相关算法

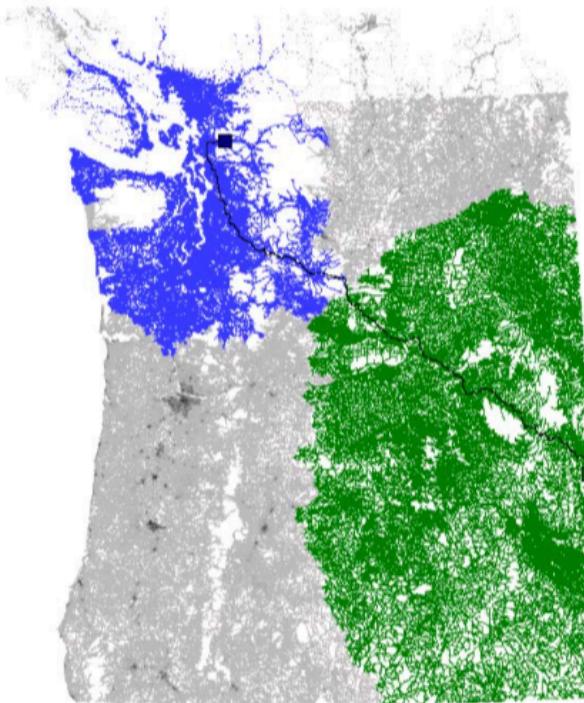
- ▶ [1959] Dijkstra最短路算法
 - ▶ 起源
- ▶ [~1968] Heuristic A*搜索算法
 - ▶ 结合实际信息
- ▶ [1991, 2002] Heuristic Hierarchical Approaches
 - ▶ 层次化
- ▶ [2005] Highway Hierarchies (HH)
 - ▶ 层次化的精确解
- ▶ [2005] Highway-Node Routing (HNR)
 - ▶ 层次化的一般抽象
- ▶ [2008] Contraction Hierarchies (CH)

方法 PathFinding.js

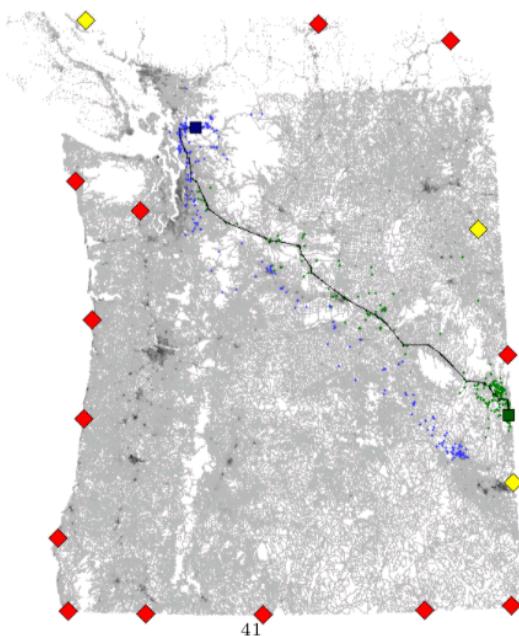
Dijkstra's Algorithm



Bidirectional Dijkstra's Algorithm



Reaches and Landmarks (with Shortcuts)



时间

- 查询
- 预处理
- 关注点
- Dijkstra 1s
- CH 0.1ms
- 好像还有更快的

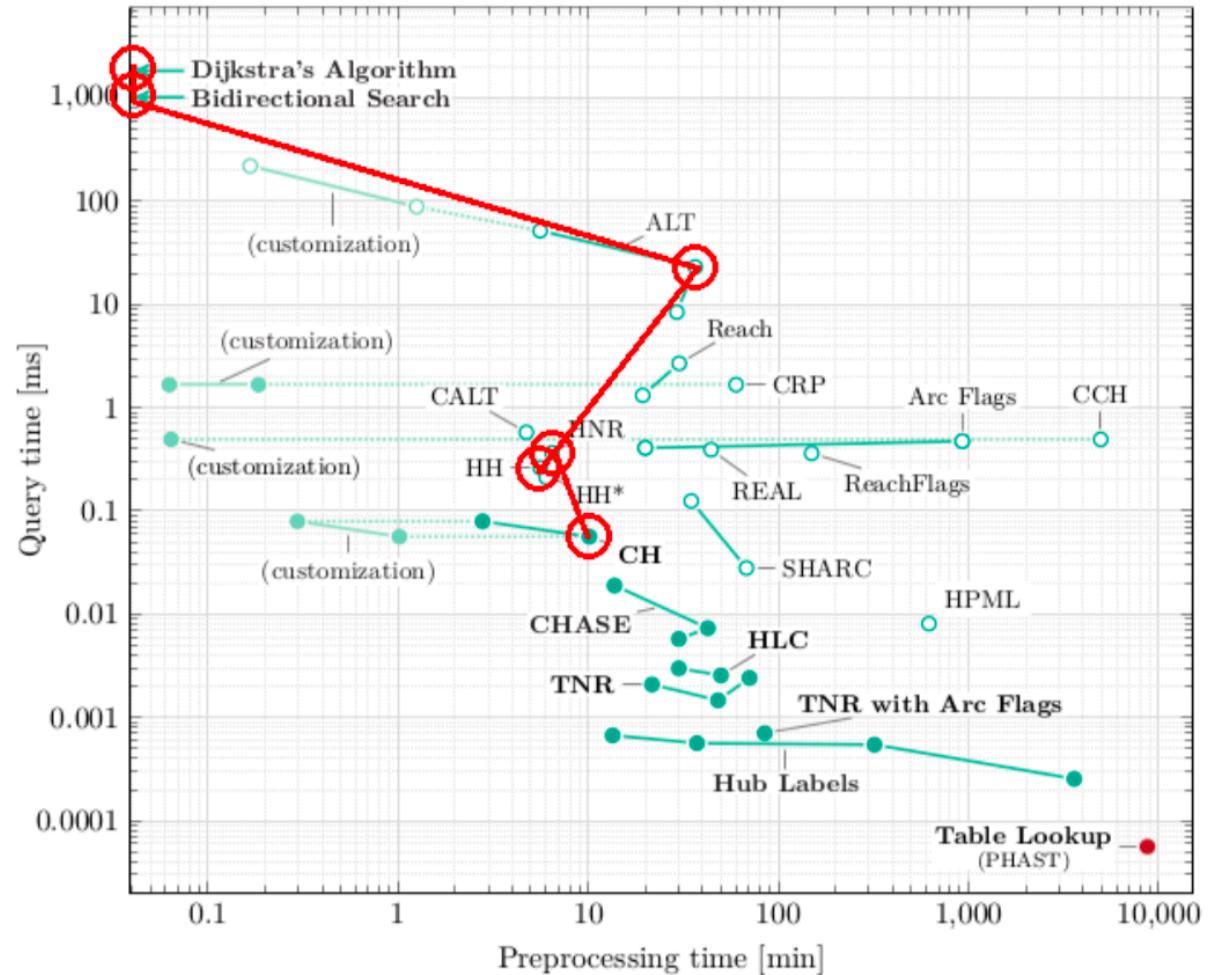


Table lookup

- 计算全部点之间的距离和路线
- 预处理花销：10000min 复杂度
- 查询：快
- 存储花销：（距离） $1.6M * 1.6M$

- 更关键的问题
- 道路不是不变的（限时通过）
- 路线不止一种（路程最短、时间最短）



总结

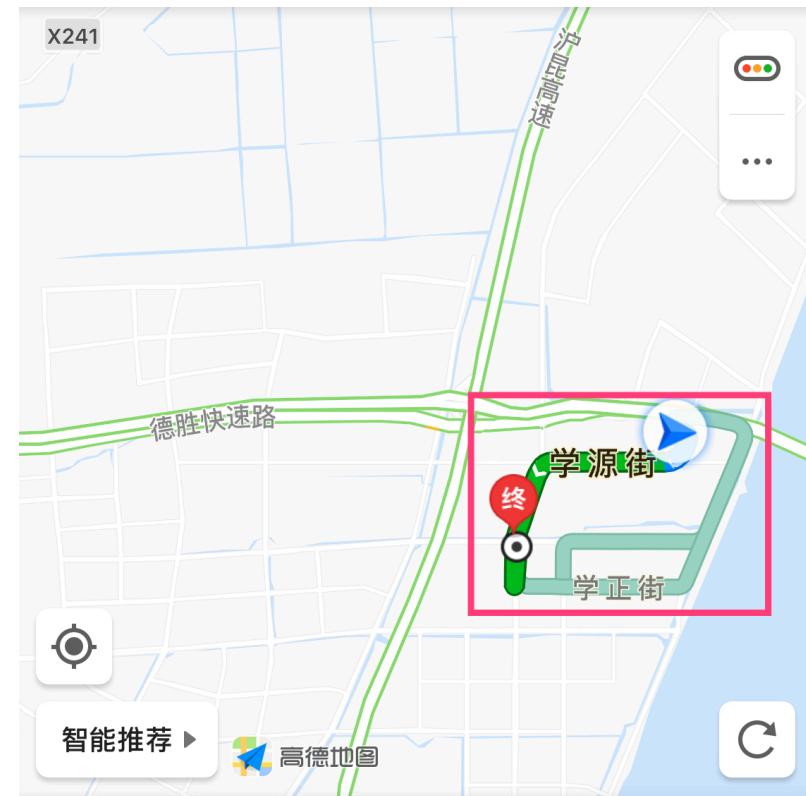
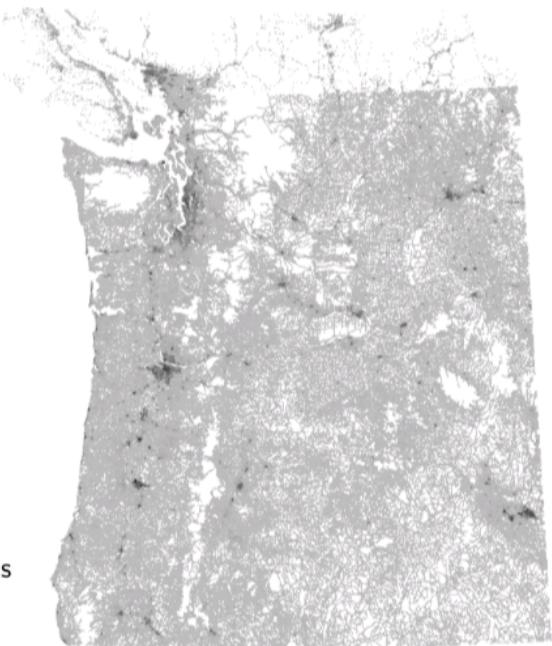
- 同样复杂的问题，竟然可以想到即减少时间又减少空间的算法，代价仅仅是认真思考而已。

路径规划

- 问题：给定起点和终点，返回一条**最优**的路线序列。
- 1. 子路网划分
- 2. 寻路（最多不超过200）
- 3. 排序（返回一种方案）
- 优化

子路网划分

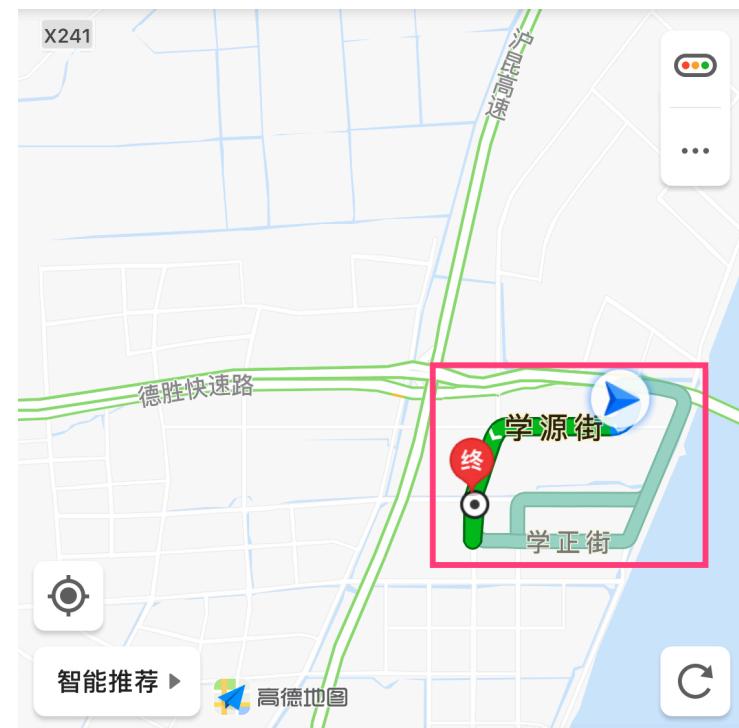
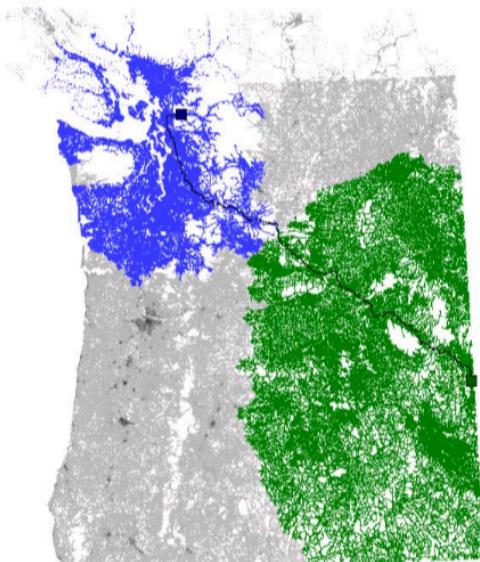
orthwest
= 1.6M vertices
 $\iota = 3.8M$ arcs



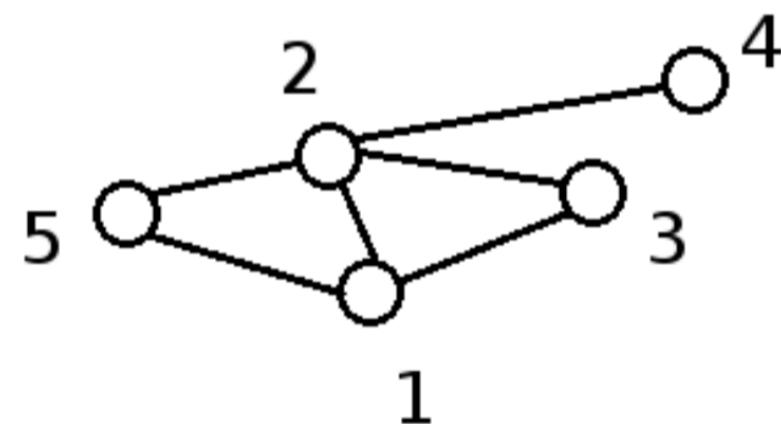
子路网划分

保证最短路包含在子路网中

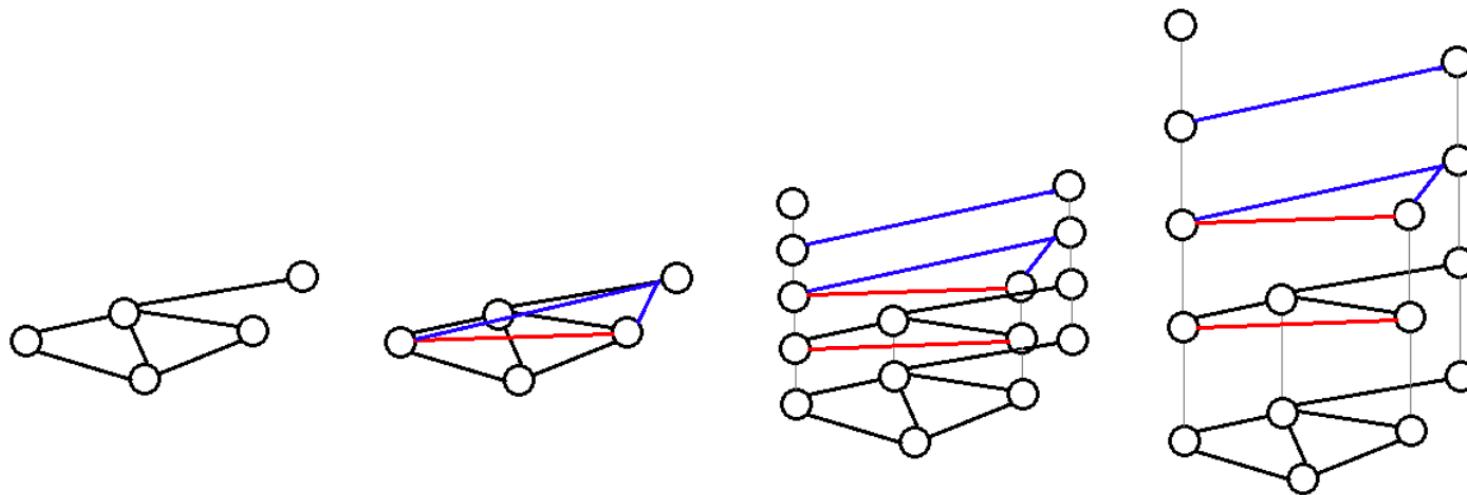
Bidirectional Dijkstra's Algorithm



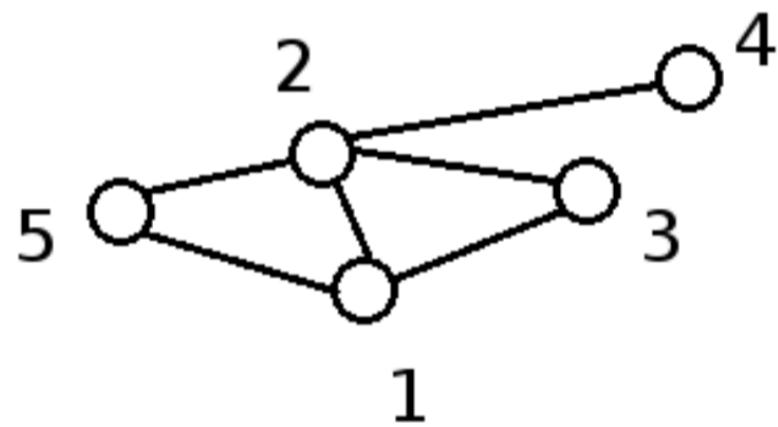
CH 算法



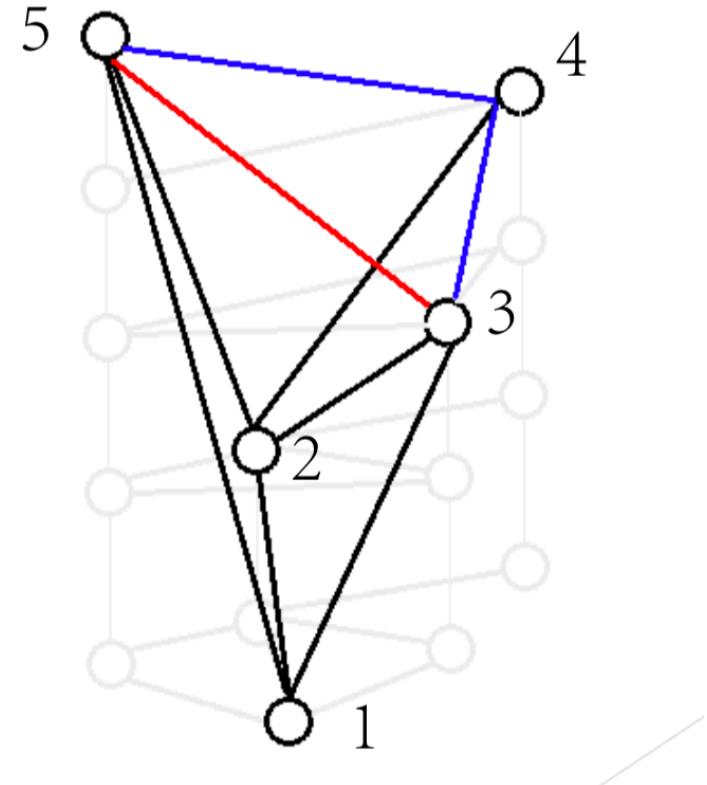
CH 算法



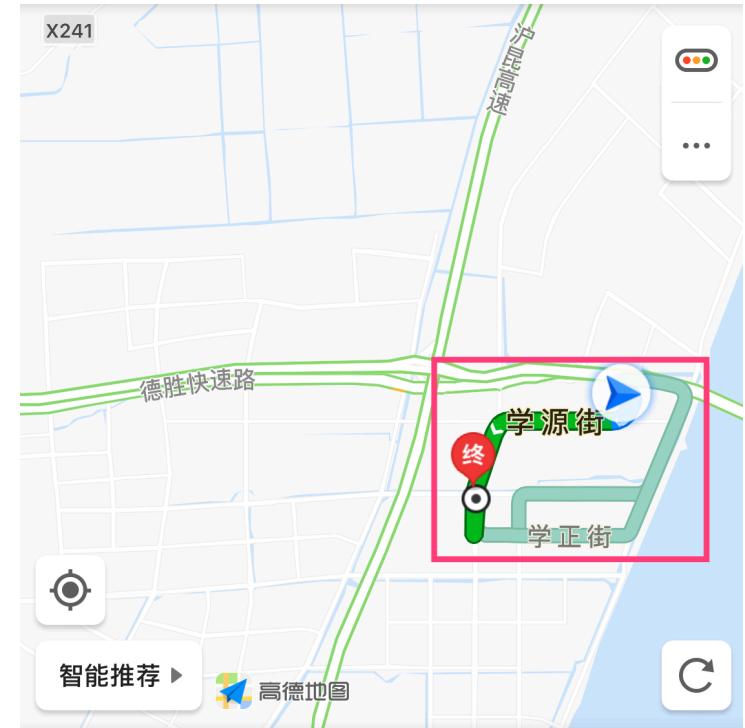
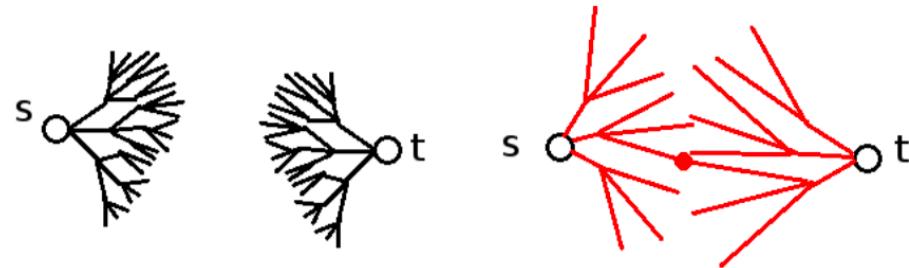
CH 算法



1. 将图转换为n层的立体模型，每个点的高度不同
2. 两个点添加一些辅助边，实现最短过程中，只需要向上搜索



CH 算法

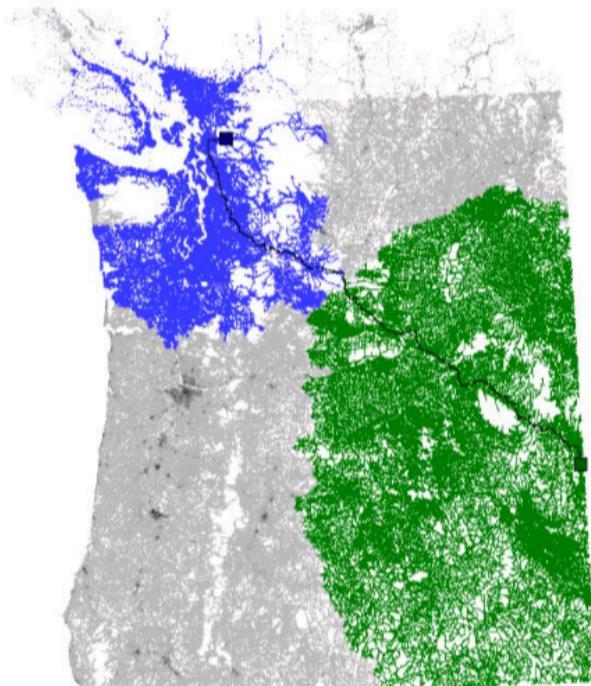


- 起点和终点有不同的高度，所以一定在更高的层相遇，记录相遇时的高度 h
- 从起点和终点分别搜索 h 的高度，然后记录下所有经过的点，组成子图

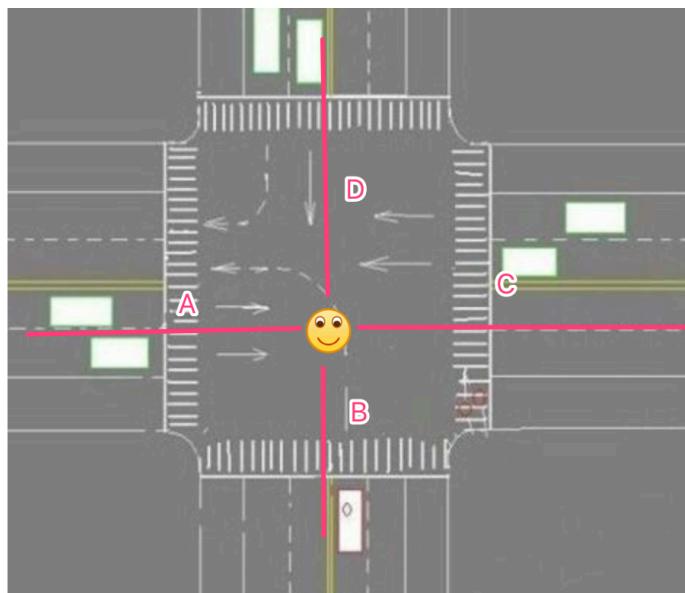
寻路

- 获得200条路线（时间权）
- 不停运行双向算法
- 直到相遇200次
- 优化：分层

Bidirectional Dijkstra's Algorithm



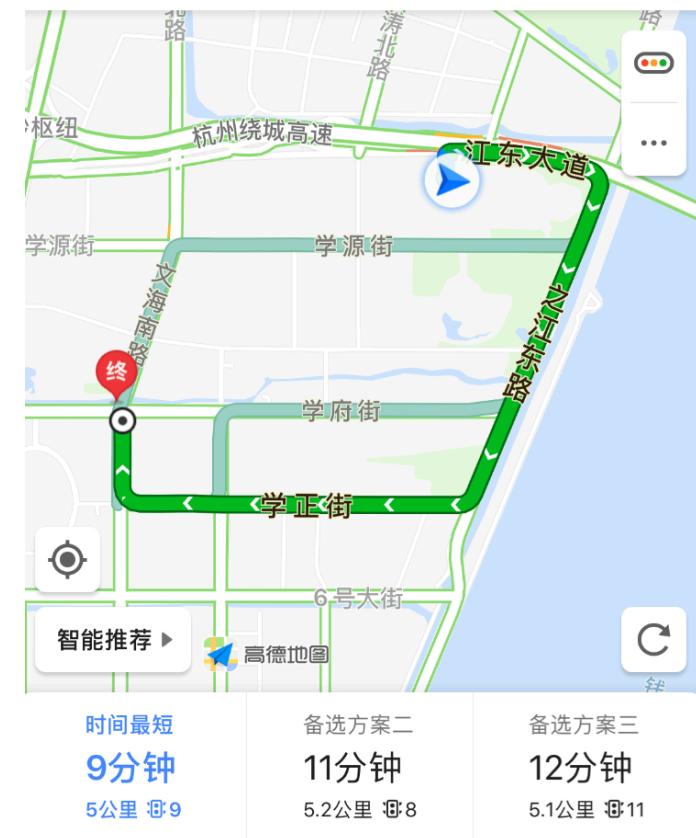
现实挑战：交通规则



- 路线其实是一个序列比如 (1-2-3-4-5-6)
- 交通规则也能转换成为一个序列 (A-B)
- 转换成为一个字符串匹配问题
- 两个序列：KMP、暴力
- 多个查询：AC 自动机

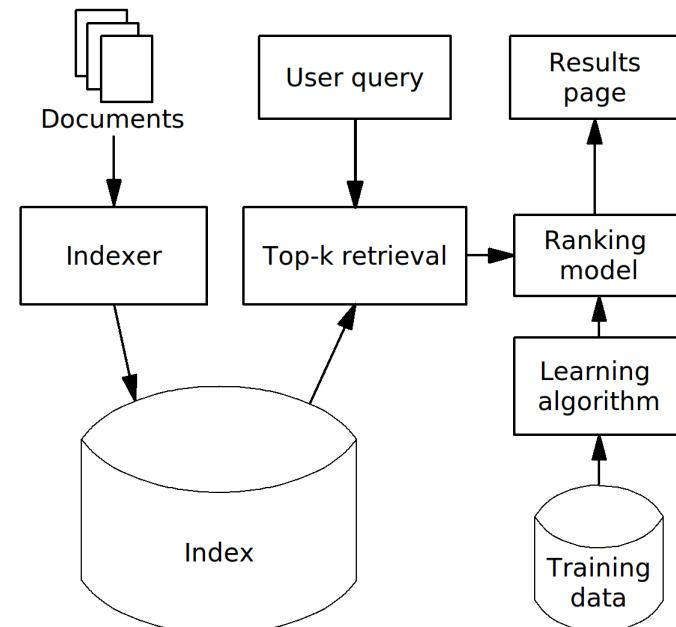
排序

- 在上一步求出的方案中，选择一个最优的结果，并且返回。
- 考虑：距离、时间、红路灯数、拐弯数、道路类型

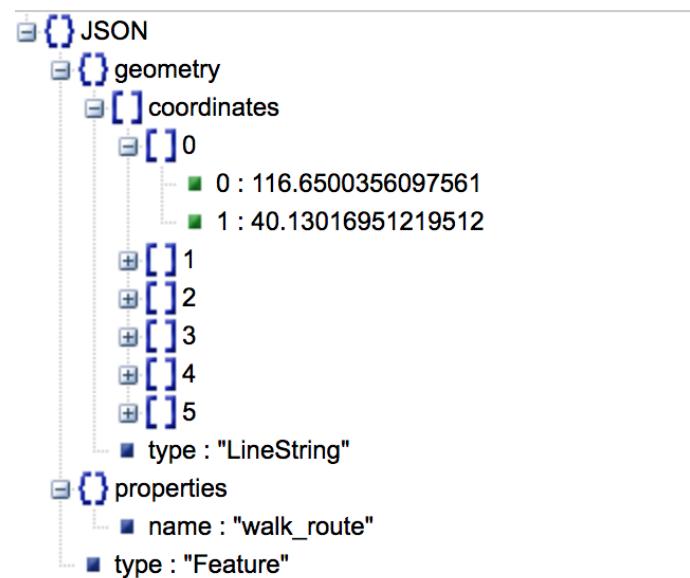


排序

- 最简单：给定一个数组，从小到大排序。
- 复杂一点：利用 $f(x) = w_1*x_1+w_2*x_2+\dots+w_n*x_n$ 计算一个值，按值排序
- 更复杂：机器学习、深度学习相关模型以及算法：Learn to rank



最终结果



```
{  
  "geometry": {  
    "coordinates": [  
      [  
        116.6500356097561,  
        40.13016951219512  
      ],  
      [  
        116.650050,  
        40.130040  
      ],  
      [  
        116.650050,  
        40.130040  
      ],  
      [  
        116.650980,  
        40.130040  
      ],  
      [  
        116.650980,  
        40.130040  
      ],  
      [  
        116.650080,  
        40.130040  
      ]  
    ],  
    "type": "LineString"  
},  
  "properties": {  
    "name": "walk_route"  
},  
  "type": "Feature"  

```

最终结果

路径规划查询工具 工具箱 图层控制

Route Plan (选择route broker或者route plan或者多路线查询)

route plan服务配置

map-rpoffline-mrp04.xg 9090 添加IP和端口

线上:ip:map-traffic-routeplan00.gz01,port:9090 删除

plan测试0:ip:map-rpoffline-mrp04.xg01,port:9090 删除

route plan生成规划列表

116.410288,39.995337 116.40992,39.99514

route plan

① 116.410288,39.995337 116.40992,39.99514

最短时间 最短距离

线上: 距离:1260米,时间:null秒

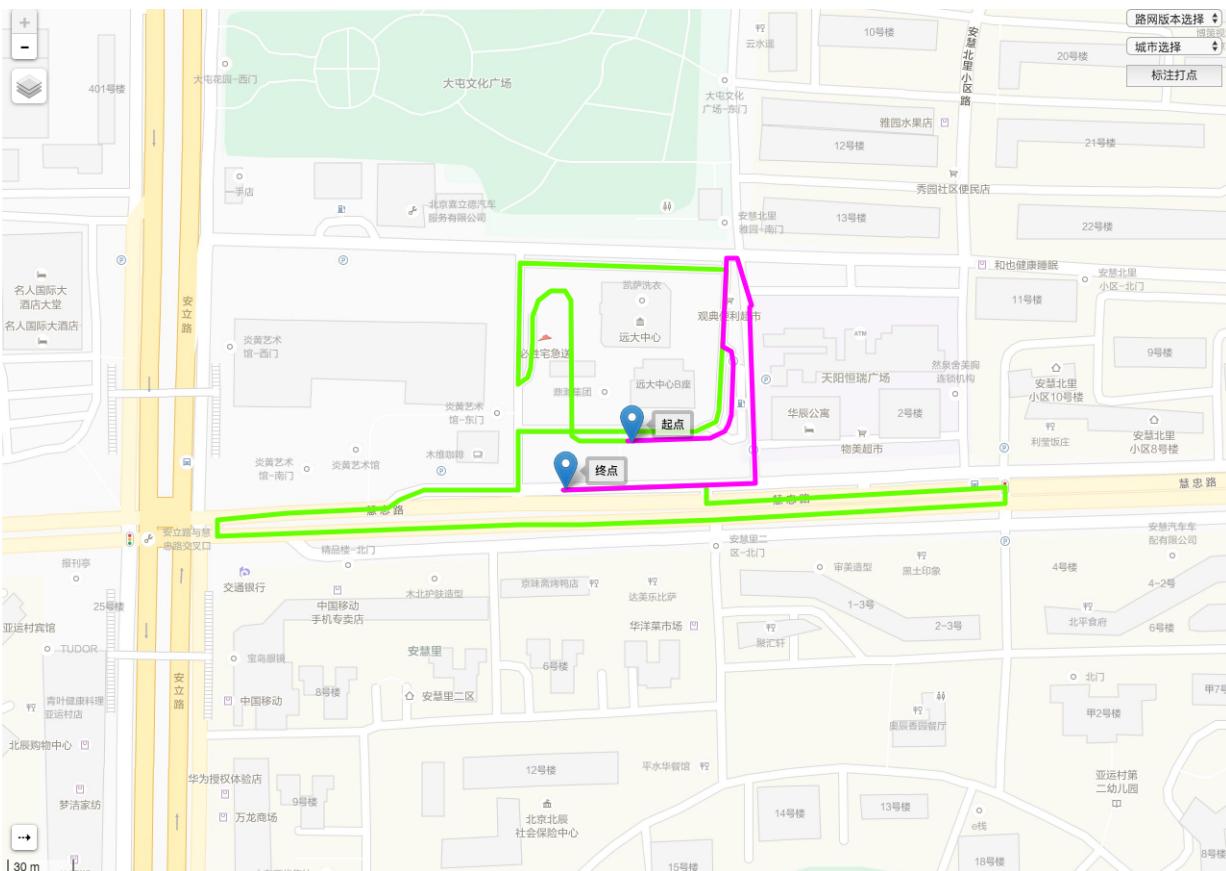
高德: 距离:32米,时间:1秒

plan测试0: 距离:334米,时间:null秒

腾讯: 距离:841米,时间:172秒

百度: 距离:931米,时间:212秒

30 m



评价

- 子路网召回率
- TOP1、TOP3
- 偏航率：司机实际行驶的轨迹和路径规划的轨迹的相似程度。
35%

优化

- 计算机理论优化

```
1 /*square1.cpp*/
2 #include <iostream>
3 using namespace std;
4 #define N 1000
5 int a[N][N] = {0}, b[N][N] = {0}, c[N][N] = {0};
6 int main() {
7     int i, j, k;
8     for (i = 0; i < N; i++) {
9         for (j = 0; j < N; j++) {
10            a[i][j] = i+j;
11            b[i][j] = i+j;
12        }
13    }
14    for (i = 0; i < N; i++) {
15        for (j = 0; j < N; j++) {
16            for (k = 0; k < N; k++) {
17                c[i][j] += a[i][k] * b[k][j];
18            }
19        }
20    }
21 }
```

```
1 /*square2.cpp*/
2 #include <iostream>
3 using namespace std;
4 #define N 1000
5 int a[N][N] = {0}, b[N][N] = {0}, c[N][N] = {0};
6 int main() {
7     int i, j, k;
8     for (i = 0; i < N; i++) {
9         for (j = 0; j < N; j++) {
10            a[i][j] = i+j;
11            b[i][j] = i+j;
12        }
13    }
14    for (i = 0; i < N; i++) {
15        for (k = 0; k < N; k++) {
16            for (j = 0; j < N; j++) {
17                c[i][j] += a[i][k] * b[k][j];
18            }
19        }
20    }
21 }
```

大学感悟

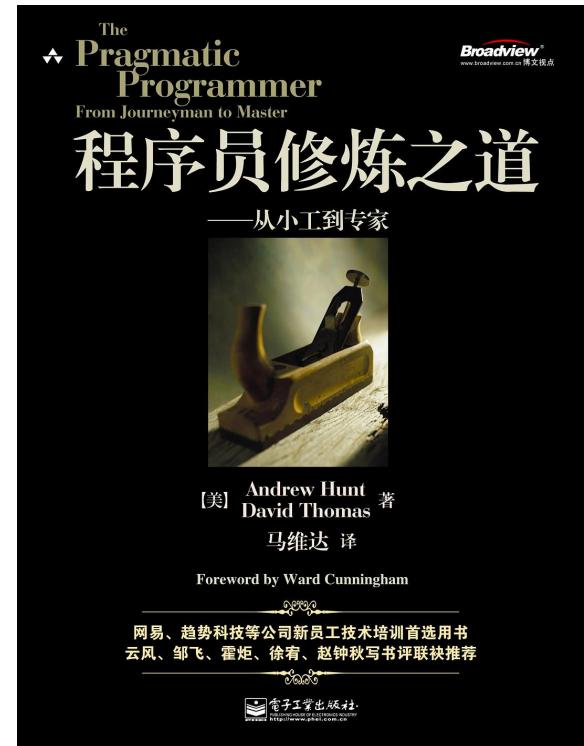
- 一门程序设计语言
- 一个开发工具
- 一个系统

程序设计语言

- C++ Java Python ...
- 程度：基础+高级语法
- 最佳：发展情况 C++11/C++14/C++17/C++20

开发工具

- 《程序员修炼之道》
- 掌握一个编辑器：Emacs、Vim
- IDE：Visual Studio、JetBrain
- 脚本语言、环境配置、软件开发流程



系统

- Windows/Linux(Ubuntu)
- 理解自由软件精神，参与开源社区



lzjqsdd

lzjqsdd

Love Technology, Love Life

Follow

Block or report user

UCAS-ICT

Beijing, Haidian

lzj7179@163.com

<http://www.lzjqsdd.com>

Overview Repositories 15 Stars 812 Followers 117 Following 68

Pinned repositories

scikit-learn-doc-cn

scikit-learn机器学习库中文文档翻译项目

HTML ★ 658 292

NewsSpider

爬取今日头条，网易，腾讯等新闻，并建立简单的搜索引擎

Python ★ 184 140

WifiAssist

The easy way to use Wifi through PC.

Makefile ★ 78 13

DDict

Linux有道翻译客户端（划词翻译）

Python ★ 6 1

Crack-Coding-Interview

This Repository contains all my codes for preparing interview.

Jupyter Notebook ★ 4 1

DB-Sim

A data driven and queue-based simulation tool for traffic

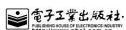
C++ ★ 1

推荐：图书

献给还在大学里的师弟师妹们，尤其是今后想干IT这一行的一群IT小小鸟，来自十几所院校，或科班，或非科班；或男生，或女生，在失望中追求希望，在迷茫中辨别方向

祝愿亲爱的师弟师妹们

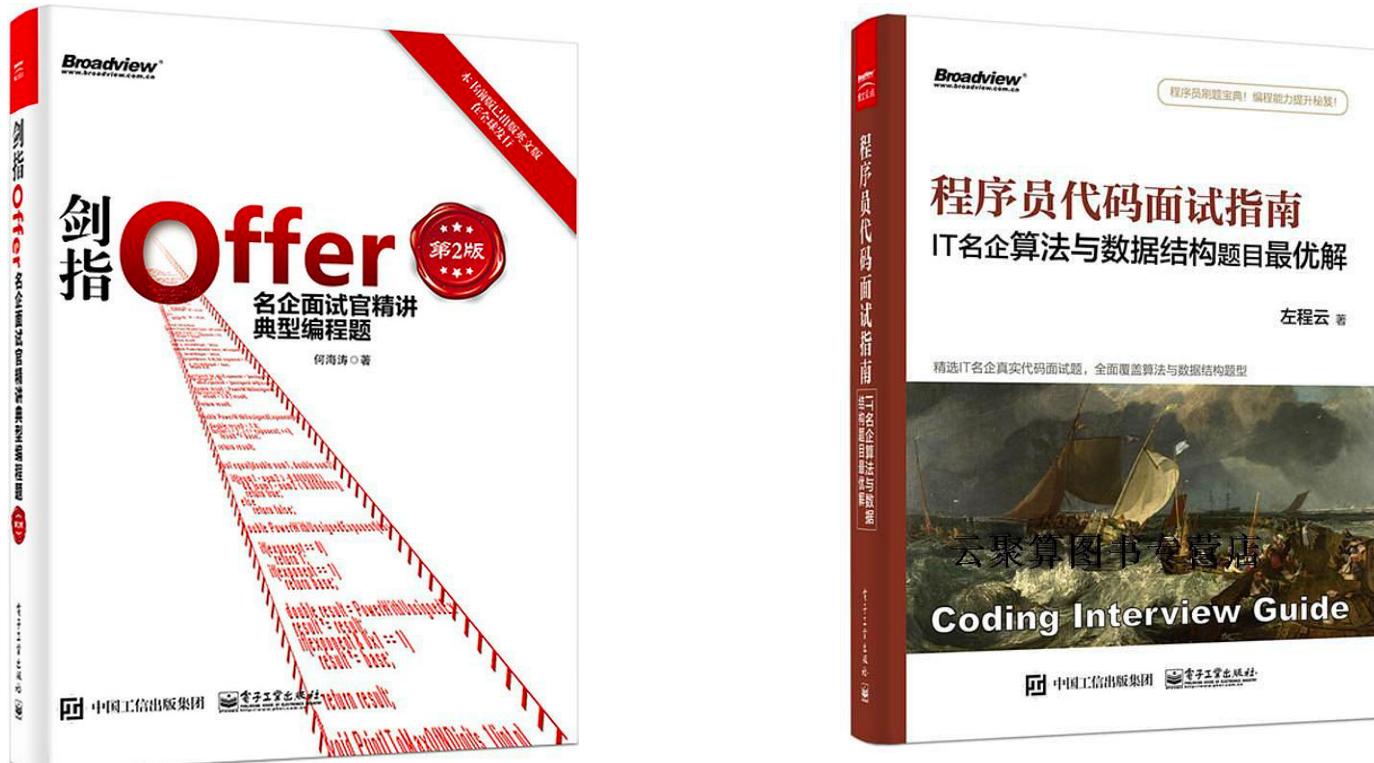
本科不虚度，考研少盲目；实习别耽误，求职心中有数
毕业≠失业，自己给自己做主



主编：胡江堂 副主编：李成 唐雅薇 秦琴



算法



博客

- 刘未鹏 Mind Hacks <http://mindhacks.cn/>
- 张驰原 Free Mind <http://freemind.pluskid.org/>
- 写你自己的博客或者 github
 - 学习记录（总结、项目代码）
 - 刻意练习（反馈途径）
 - 搜索展示

职业规划

- V2EX - 酷工作
- 牛客网 - 讨论区（应届生实习和校招）
- 脉脉 - 匿名区（程序员工资有多高、业内风向）
- 简历、面试

Q&A

- 联系方式
- QQ : 523689985
- 微信 : RickXiang
- 博客 : xiang578.com

参考链接

- 如何看待滴滴章文嵩称，滴滴的算法比 AlphaGo 复杂 100 倍？ - 知
<https://www.zhihu.com/question/272009611/answer/365868288>
- 滴滴文嵩的微博_微博
https://weibo.com/wensong8?profile_ftype=1&is_all=1#_0
- 最短路演示 <https://qiao.github.io/PathFinding.js/visual/>
- 最短路相关资料
<http://www.cs.princeton.edu/courses/archive/spr06/cos423/Handouts/EPP%20shortest%20path%20algorithms.pdf>
- Vim <https://github.com/spf13/spf13-vim>
- Vim <https://github.com/SpaceVim/SpaceVim>
- Emacs <https://github.com/syl20bnr/spacemacs>