Chapter0. Introduction

2019年2月20日

18:59

Note Taking Area

**Course information**

* Textbook: Algorithm design.
* Grades:
  + Final exam 40%
  + Lab 30%
  + Homework 20%
  + Attendance 10% - Be aware that sign in will be carried in lecture.
* Office hours: Wednesday 14:00 - 16:30, room 1011 A7 building, Zhiyuan.
* The lecture teacher has heavy accent.

Cue Column

Summaries

1. Course information.

Chapter1. Some Representative Problems: G-S

2019年2月20日

19:28

Note Taking Area

**Stable matching**

* 雇主与雇员的两条优先队列，满足稳定情况问题。
  + 稳定情况（其中之一成立）：
    1. 雇主E对他所接受的每个雇员都对该雇员A感兴趣；
    2. 雇员A对他目前工作状况比雇主E感兴趣。
  + Unstable condition (both hold):

Unstable pair: applicant x and hospital y are unstable if: 
. x prefers y to its assigned hospital. 
. y prefers x to one of its admitted students. 

* **Stable assignment**: assignment with no unstable pair.
* Simplify into base case: one to one match, and all in match.
  + We'll consider the marriage as example in the following section.

**Gale-Shapley algorithm / propose-and-reject algorithm**

* The description of steps of algorithm:
  1. A male M make a proposal to highest female W, they are in date.
  2. Then every step the male A whole hasn't in date make a proposal to a highest priority female whole hasn't been made a proposal by A, and female selects the males and choose the highest priority (the other one isn't in date again).
  3. Until all males and females are in date, the algorithm done.
* The pseudo code:

初 始 所 有 的 m e M 和 we W 都 是 自 由 的 
While 存 在 男 人 m 是 自 由 的 且 还 没 对 每 个 女 人 都 求 过 婚 
选 择 这 样 一 个 男 人 m 
令 是 m 的 优 先 表 中 m 还 没 求 过 婚 的 最 高 排 名 的 女 人 
If w 是 自 由 的 then 
(m ， 训 ） 变 成 约 会 状 态 
Else w 当 前 与 m ' 约 会 
If w 是 更 偏 爱 m' 而 不 爱 m then 
m 保 持 自 由 
Else 训 更 偏 爱 m 而 不 爱 m ， 
（ m ， w) 变 成 约 会 状 态 
m' 变 成 自 由 
Endif 
Endif 
Endwhile 
输 出 己 约 会 对 的 集 合 5 ． 
图 1.2 
女 人 w 将 变 成 与 m 
约 会 ， 如 果 她 电 偏 
爱 他 ， 而 不 是 
O 
O 
O 
当 自 由 男 人 m 向 女 人 
w 求 婚 时 ， G-S 算 法 的 
一 个 中 间 状 态 

Cue Column

**Analysis and proof of G-S algorithm**

* Female ω keeps engagement since her first proposal, and her couple becomes better.
* 男性m求过婚的女性在优先表中越来越差。
* G-S算法至多在n2次迭代之后终止。

证 正 如 我 们 这 里 打 算 做 的 ， 一 个 给 出 算 法 运 行 时 间 上 界 的 有 用 策 略 是 找 出 进 展 的 度 
量 标 准 ． 即 我 们 找 出 某 种 精 确 的 方 法 说 这 个 算 法 执 行 的 每 一 步 使 得 它 离 终 止 更 接 近 ， 
在 当 前 这 个 算 法 的 情 况 下 ， 每 次 迭 代 由 某 个 男 人 向 一 个 以 前 没 有 求 过 婚 的 女 人 （ 唯 一 的 
一 次 ） 求 婚 构 成 ， 于 是 ， 如 果 令 P （ t) 表 示 到 迭 代 ' 结 束 m 己 经 向 “ 求 过 婚 的 那 些 （ m “ ， ） 对 的 
集 合 ， 我 们 看 到 ， 对 所 有 的 ' ， PO + I) 的 大 小 严 格 大 于 P(t ） 的 大 小 ． 但 是 总 计 只 存 在 ” 2 个 可 
能 的 男 人 和 女 人 的 对 ， 因 此 PC• ） 的 值 在 算 法 的 整 个 进 程 中 至 多 可 能 增 加 ” 2 次 ． 从 而 得 到 
至 多 可 能 存 在 定 次 迭 代 ， 

* 如果m在算法执行的某时刻是自由的，那么存在一个他还没有求婚的女性。

证 假 设 存 在 一 个 点 ， 此 刻 m 是 自 由 的 ， 但 是 已 经 向 每 个 女 人 都 求 过 婚 了 ． 那 么 根 据 命 
题 1.1 ， ” 个 女 人 中 的 每 个 人 在 这 个 时 间 点 上 是 约 会 状 态 ． 由 于 约 会 的 对 的 集 合 构 成 一 个 匹 
配 ， 在 这 个 时 间 点 上 一 定 也 有 个 约 会 的 男 人 ， 但 是 ， 总 共 只 有 ” 个 男 人 ， 且 m 不 在 约 会 ， 因 
此 这 是 矛 盾 ． 

* 终止时返回的集合S是一个完美匹配。
* G-S算法的一次执行，返回一个稳定匹配集合S。
* 所有的执行得到同样的匹配，G-S算法的每次执行都得到集合S\*，其中S\*表示一个对的集合。
* 在稳定匹配S\*中每位女性都与她最差的有效伴侣配对。

**Lesson submission and score**

* The online lab submission: 30%
  + Only one submission one week, which contains 10 test cases and each test case has 10 points.
  + The first submit will get full score, and one wrong answer will deduct 5 points.
* The theorical homework: 20%
  + Which will be submitted on sakai, at irregular intervals.
* Plagiarism policy:
  + If you use online code, write comment of original website.
* Code standards:
  + 《阿里巴巴Java开发手册 1.3.1》
  + 《Google style guide》
    - <https://google.github.io/styleguide/>

Summaries

1. Stable matching.
2. Gale-Shapley algorithm.

Chapter3. 图

2019年3月13日

19:41

**NOTE TAKING AREA**

基本定义与应用

* **命题3.1** 每棵**n个节点的树**恰好**有n-1条边**。
* **定理3.2** 设G是具有**n个节点的无向图**，下面**任意两个语句**都可以**推出第三个语句**：
  + G是**连通**的。
  + G**不包含一个圈**。
  + G有**n-1**条边。

宽度优先搜索

* 连通性问题（s-t连通性的确定问题）：在G中是否存在一条从s到t的路径？
* BFS算法构造的层：
  + 层L1由s的所有邻居组成（有时也用L0表示）。
  + 对于未定义的第一个层Lj+1由不属于前面的层并且有一条边通向层Lj的某个节点的所有节点组成。
* **定理3.3** 对每个j≥1，由BFS产生的层Lj恰好由所有到s距离为j的节点组成，存在一条从s到t的路径当且仅当t出现在某个层中。
* **定理3.4** 设T是一棵宽度优先搜索树，设x和y是T中分别属于层Li和Lj的节点，并且设(x,y)是G的一条边，那么i与j至多差1。
* 使用BFS确定连通分支：

R 些 有 一 路 径 从 、 向 它 的 
While 存 一 ( ) , u€-RfiveR 
, , 加 刊 R 中 
End 、 vhilc 

* 算法结束时产生的集合R恰好是G的包含s的连通分支。

深度优先搜索

* 深度优先搜索（DFS）的伪代码：

DFS( , 
11 v % "Explored"'hen 
DFS(v) 
Endii 

* 命题和定理：
  + **命题3.6** 对于给定的递归调用DFS(u)，在这次激活和这个递归调用结束之间被标记为“Exprored”的所有的节点都是u在T中的后代。
  + **定理3.7** 设T是一棵深度优先搜索树，x与y是T中的节点，且(x,y)是G中不属于T的一条边，那么x或y之中一个是另一个的祖先。
  + **定理3.8** 对图中任两个节点s与t，它们的连通分支或者相等，或者不相交。

用优先队列与栈实现图的遍历

* 图的表示：邻接表和邻接矩阵。
  + C:\F52AA0E5\728ABA0C-5013-449A-B6B8-299C3DB913FF.files\image007.png
    - 其中m为与v相交的边数，Adj[v]的表长是nv。
  + **定理3.10** 一个图的邻接矩阵表示需要O(n2)的空间，而邻接表表示只需要O(m+n)的空间。
* 宽度优先搜索的实现：

- な ご を 言 物 物 
エ ・ ト を 崟 1 ・ ー ) 製 
「 こ 7 ↓ 第 、 キ 物 ま こ 
上 上 , を 3 
上 宿 

* **定理3.11** 如果图是由邻接表给出的，BFS算法的上述实现将以O(m+n)时间复杂度运行。
* 深度优先搜索的实现

OFS(s) 
while S 
W ExploredCuJ—faIse then 
true 
For 
End for 
Endwhile 

* 考虑将Explored标记放置于元素入栈之前，以节约内存占用。
* **定理3.12** 上述算法在下面的意义上实现DFS，它按照与上一节递归的DFS过程恰好相同的次序（除了每个邻接表是按照相反的次序处理之外）访问节点。
* **定理3.13** 如果图是由邻接表给出的，DFS算法的实现将以O(m+n)时间运行。

二分性测试：宽度优先搜索的应用

* 二部图：对结点着红色和蓝色，使得每条边有一个红端点和一个蓝端点。
* **定理3.14** 如果一个图是二部图，那么它不可能包含一个奇圈。
* **定理3.15** 设G是一个连通图，令L1，L2，…是由始于结点s的BFS所产生的层，那么下面两件事一定恰好成立其一：
  + G中没有边与同一层的两个节点相交，在这种情况下G是二部图，其中偶数层的节点可以着红色，奇数层的节点可以着蓝色。
  + G中有一条边与第一层的两个节点相交。在这种情况下，G包含一个奇数长度的圈，且因此不可能是二部图。

有向图中的连通性

* **命题3.16** 如果u和v是相互可达的，v和w是相互可达的，那么u和w也是相互可达的。
* **定理3.17** 对有向图中的任何两个节点s和t，它们的强连通分支或者相等，或者不相交。

*DAG与拓扑排序的内容省略，请参考DSAA相关内容。*

**CUE COLUMN**

**SUMMARIES**

1. 图的基本定义和应用。
2. 宽度优先搜索。
3. 广度优先搜索。
4. 用队列和栈实现图的遍历。
5. 二分性测试：宽度优先搜索的应用。
6. 有向图的连通性。

Chapter4. 贪心算法

2019年3月20日

14:18

**NOTE TAKING AREA**

贪心算法

* 定义：算法通过一些**小的步骤**来建立一个解，在每一步根据**局部情况**选择一个决定使得某些**主要的指标能得到优化**。
* 贪心算法领先：贪心算法的进展**每一步**都比其他算法好。
* 交换论证：考虑对某问题的任何可能的解，逐渐把它**转换成由贪心算法找到的解而不损害它的质量**。
* 例题：图中的最短路径、最小生成树问题、Huffman码，最小费用有向树问题。

区间调度：贪心算法领先

* 一点一个需求i1被接受，则拒绝所有与i1不相容的需求，选择下一个被接收的需求i2，并且拒绝与i2不相容的需求，直到遍历完所有需求。

始 令 R 是 所 需 的 集 合 ． 设 为 空 
WhileR-f•æ 
择 一 个 有 最 小 車 耐 同 隶 i 《 
把 到 」 中 
从 R 中 除 与 葛 裂 不 客 的 所 有 製 
Endwhile 
返 0 台 A 作 为 被 的 莴 求 集 合 

* 分析算法：首先，算法返回的集合A中的区间都是相容的。
  + **命题4.1** A是一个相容的需求集
  + C:\F52AA0E5\728ABA0C-5013-449A-B6B8-299C3DB913FF.files\image011.png
  + **命题4.3** 贪心算法返回一个最优的集合A。

证 我 们 现 在 用 反 证 法 证 明 这 个 论 断 ． 如 果 A 不 是 最 优 的 ， 那 么 一 个 最 优 集 合 0 一 定 
有 更 多 的 需 求 ， 即 我 们 有 一 定 有 m > k. 对 ， = 玉 应 用 命 题 能 2 ， 我 们 得 到 不 孙 〕 < 不 ． 因 为 
m > 在 0 中 存 在 一 个 需 求 j “ 吓 这 个 需 求 在 结 束 之 后 丌 始 ， 因 此 也 在 结 束 之 后 廾 始 、 
所 以 在 删 除 了 所 有 与 需 求 0 癖 ， ． “ 不 相 容 的 需 求 之 后 ， 可 能 的 需 隶 集 合 R 仍 旧 包 含 丿 ， 亠 
但 是 贪 心 算 法 i. 停 十 在 孙 ， 而 假 设 它 仅 当 为 空 时 停 止 ， 产 生 矛 盾 ． 

* 实现与运行时间：算法运行时间为O(nlogn)。
* 相关问题：调度所有的区间

间 题 在 区 间 调 度 问 题 中 ， 存 在 一 种 单 一 的 资 源 和 以 时 间 区 间 形 式 表 示 的 许 多 需 求 ， 因 
此 我 们 必 须 诜 择 哪 糕 需 求 被 接 受 且 嚦 些 需 求 被 拒 绝 ． 如 果 我 们 有 许 多 相 同 的 资 源 可 用 ． 面 且 
我 们 想 使 用 尽 可 能 少 的 资 源 安 排 所 有 的 需 求 ， 那 么 就 产 生 一 个 相 关 的 同 题 ． 由 于 这 里 的 目 标 
是 把 所 有 的 区 间 划 分 到 多 个 资 源 ， 我 们 将 把 它 看 做 一 个 区 间 划 分 同 题 ． 
例 如 ， 假 设 每 个 求 与 一 个 在 特 定 时 同 区 间 教 室 里 需 要 安 排 的 课 对 应 ． 我 们 想 使 用 尽 可 
能 少 的 教 室 满 足 所 有 这 些 需 求 、 因 而 在 我 们 的 安 排 下 教 室 是 多 个 资 源 ， 并 且 基 本 的 约 束 是 两 
个 在 时 间 上 重 叠 的 课 必 须 被 安 排 在 不 同 的 教 室 ． 采 用 等 价 的 说 法 ， 区 间 需 求 也 可 以 是 需 要 在 
特 定 时 间 区 间 被 处 理 的 工 作 ， 而 资 源 可 能 是 能 够 处 理 这 些 工 作 的 机 器 ． 

* **命题4.4** 在任何区间划分的实例中，资源数必须至少是区间集合的深度。
* 设计算法：设d是区间集合的深度，显示怎样对每个区间分配一个*标签*，即数集{1, 2, …, d}，重叠的区间用不同的数做标签（名字对于数字的映射）。

任 意 打 破 区 同 的 身 列 ． 根 据 弄 始 對 回 对 它 们 序 
“ ， I. 表 示 按 这 个 次 序 排 列 为 区 甸 
For 每 个 按 上 述 次 序 瓴 先 于 并 且 与 它 童 叠 区 司 不 
从 对 的 考 内 中 除 I, 杓 标 簍 
Endfor 
If 在 { 1, 六 一 ， 中 存 在 饪 伺 还 没 有 被 排 肇 为 标 莶 then 
分 配 一 个 排 的 朽 筌 给 
俣 不 分 配 标 
Endfor 

* **命题4.5** 如果我们使用上述贪心算法，每个区间将被分配一个标签，且没有两个重叠的区间接受同样的标签。
* **定理4.6** 上述贪心算法使用与区间集合深度等量的资源为每个区间安排一个资源，这是所需资源的最优数量。

最小延迟调度：一个交换论证

再 次 考 虑 下 述 情 况 ， 我 们 有 单 一 资 源 和 一 组 使 用 资 源 的 ” 个 需 求 ， 每 个 需 求 需 要 一 个 时 
间 区 间 ． 假 定 资 源 在 时 刻 、 开 始 有 效 ． 但 是 与 前 面 的 问 以 相 反 ， 每 个 需 求 i 骁 在 是 更 加 灵 活 ． 
不 是 开 始 时 间 与 结 束 时 回 ， 需 求 i 有 一 个 截 ] k 时 间 出 ， 并 且 它 要 求 一 个 长 为 的 连 续 的 时 间 
区 间 ， 但 是 它 想 被 安 排 在 截 止 时 间 之 前 的 任 何 时 刻 ， 每 个 被 接 受 的 需 求 必 须 被 分 配 一 个 长 为 
《 的 时 间 区 间 ， 且 不 同 的 需 求 被 分 在 不 重 叠 的 区 同 ． 

* 具有最小截止时间的任务优先排列：

怙 郦 它 纩 的 裁 让 时 同 蚋 序 对 任 务 排 序 
为 身 号 的 单 ， 假 定 山 < 一 •<d, 
按 这 个 冰 序 考 任 一 l. 六 
什 务 ' 女 排 在 从 、 00 。 7 到 丿 行 ， 一 + 过 ， 的 时 回 区 回 
对 一 1.2 ， “ ， “ 》 近 河 排 的 这 塑 区 同 [ i) ， 不 i) 〕 

* 空闲时间：机器停工了但是还有任务没有安排，称为空闲时间。
  + 存在一个没有空闲时间的最优调度。
  + 所有没有逆序也没有空闲时间的调度有相同的最大延迟。
    - 逆序：截止时间较晚的任务安排在较早截止时间的前面。
  + 存在一个既没有逆序也没有空闲时间的最优的调度。
  + 由这个贪心算法产生的调度A有最优的最大延迟。

最优超高速缓存：一个更复杂的交换论证

当 处 理 存 储 分 层 的 时 候 ， 恰 好 会 产 生 这 个 问 题 ． 有 少 量 数 据 可 以 被 快 速 存 取 ， 而 大 量 数 
据 需 要 更 多 的 时 间 存 取 ； 你 必 须 决 定 哪 些 数 据 块 近 在 手 边 ． 

* 超高速缓存：在一个快速存储器中存储少量数据以便减少与一个慢速存储器的交互而花费的时间。

When イ 書 を 被 ま 入 高 達 与 
在 量 名 第 物 被 を 竹 第 を 

* 最远将来规划将产生一个最优调度S。
* 简化调度：做最少的给定步所必须的工作。设S可能不是一个简化调度，令S‘为S的简化，在任何一步S放入一个非必需的项d，而S’不将d移入超高速缓存。
  + S‘是一个与调度S放入至多一样多个项的简化调度。
  + 对于某个数j，令S是在序列中的前j项与SFF做同样收回决定的简化调度，那么存在一个在序列中的前j+1项与SFF做同样收回决定的简化调度S‘，并且S’不比S产生更多的缺失。
  + SFF比任何其他的调度S‘不产生更多的缺失，因此是最优的。

图的最短路径

最 短 路 径 问 题 的 具 体 结 构 如 下 ： 给 定 有 向 图 G 一 （ V ， 0 ， 以 及 一 个 指 定 的 开 始 结 点 5 ． 
假 设 ， 有 一 条 路 径 通 向 G 中 的 每 个 其 他 结 点 ． 每 条 边 e 有 一 个 长 度 I ， 冫 0 ， 表 示 在 ' 上 旅 行 
用 的 时 间 〔 或 者 № 离 ， 或 者 费 用 ). 对 于 路 P · P 的 长 度 一 一 用 I （ P ） 表 示 一 一 是 尹 中 所 有 边 
的 长 度 之 和 ， 我 们 的 目 标 是 确 定 从 ， 到 图 中 每 个 其 他 结 点 的 最 短 路 径 ． 我 们 应 该 提 到 ， 尽 管 
同 题 是 对 有 向 图 指 定 的 ， 我 们 也 可 以 处 理 无 向 图 的 情 况 ， 只 不 过 将 每 条 长 为 的 无 向 边 ， 一 
（ u “ ， 用 两 条 每 条 长 度 都 为 L. 的 有 向 边 （ 如 和 （ t ， ， 代 替 就 可 以 了 ． 

* Dijkstra算法：从已确认最短路径的点出发，抵达新节点的路径，

Diikatra 
White sŕv 

* 考虑在算法执行中任意一点的集合S，对每个u∈S，路径Pu是最短的s-u路径。
* 使用优先队列，Dijkstra算法可以在具有n个节点和m条边的图上实现，它运行在O(m)时间，再加上n次ExtractMin操作和m次ChangeKey操作的时间，总时间为O(mlogn).

最小生成树问题

对 于 确 定 的 对 (t 吓 的 我 们 可 能 以 某 个 费 用 以 珥 ， 10 > 0 建 立 t' ， 与 丐 之 回 的 直 接 彘 接 ． 
于 是 我 们 可 以 用 一 个 图 G 一 （ V ， E ） 来 表 示 可 能 被 建 立 的 链 接 的 集 合 ． 与 每 条 边 ， 一 0 产 t") 
相 关 的 有 一 个 正 的 用 0 回 题 就 是 找 一 个 边 的 子 集 TCE 使 得 图 （ V ， T ） 是 连 通 的 ， 且 总 费 
用 （ ， 最 小 （ 我 们 将 假 设 整 个 G 是 连 通 的 ； 否 则 没 有 可 行 解 ). 

* 令T是上述定义的网络设计问题的最小费用解，那么(V, T)是一棵树。
* Kruskal算法：初始没有边，通过费用递增添加E中的边构建生成树，如果成环则跳过边e。
  + Union-Find数据结构：给定一个节点u，操作Find(u)返回包含u的集合的名字，测试节点u和v是否在同一个集合中。操作Union(A, B)合并集合A和B.
    - MakeUnionFind(S)返回集合S上的一个Union-Find数据结构，其中所有的元素都在分离的集合里，O(n).
    - Find(u)返回包含u的集合的名字，O(logn).
    - Union(A, B)将集合A和B合并成一个集合，O(logn).
  + 通过MakeUnionFind(S)构建最初的集合，对于每条边e=(u, v)被考虑时，计算Find(u)和Find(v)检查u和v属于不同的连通分支，之后使用Union(Find(u), Find(v))合并两个分量。
  + Kruskal算法在n个结点和m条边的图上可以在O(mlogn)运行时间内实现。
* Prim算法：从根节点s开始，每一步加入边费用最小的节点。
  + 在具有n个节点和m条边的图上，使用优先队列可以实现Prim算法，其运行时间为O(m)加上n次ExtractMin和m次ChangeKey操作的时间。总运行时间为O(mlogn).
* 逆向删除算法：从整个图(V, E)开始，从最大费用删除边，如果破坏了图的连通性，则跳过这条边。
* 割性质：假设所有的边的费用都是不等的，令S是任意节点子集，它既不是空集，也不等于全集V，令边e=(v, w)是一端在S中，另一端在V-S中的最小费用边，那么每棵最小生成树都包含边e.
* Kruskal算法产生G的一颗最小生成树，Prim算法产生一棵最小生成树，逆删除算法产生G的一颗最小生成树。
* 圈性质：假设所有边的费用都是不等的，令C是G中的一个圈，令边e=(v, w)是属于C的最贵的边，那么e不属于G的任何最小生成树。

**CUE COLUMN**

命题4.2的证明

证 我 们 将 用 归 纳 法 证 期 这 个 论 断 ． 对 于 ， 一 1 ， 论 断 显 然 为 真 ： 算 法 由 选 择 具 有 最 小 截 
止 时 间 的 需 求 i 》 丌 始 ， 
理 在 今 ， > l. 根 据 归 纳 假 设 ， 假 定 这 个 论 断 对 ， 一 1 为 貞 ， 我 们 将 试 图 证 明 它 对 r 也 为 
真 ． 正 如 图 3 所 示 ， 甴 妇 纳 假 设 我 们 有 不 00 <f(je-i). 为 了 使 算 法 的 第 ， 个 区 间 不 更 
早 地 结 秉 ， 它 必 须 “ 落 后 " ， 正 如 该 图 所 示 ． 有 一 个 简 单 的 理 由 使 得 它 不 可 能 发 生 ： 不 选 择 一 
个 结 束 迟 的 区 间 ， 贪 心 算 法 （ 最 坏 情 况 下 ） 总 可 以 选 择 j ， ， 十 是 完 成 了 归 纳 步 骤 ． 
我 们 可 以 更 精 确 地 如 ． 卜 表 达 这 个 论 述 ． 我 们 知 道 丿 0 冖 ） ， (j ， 以 因 为 0 由 相 容 的 区 间 
组 成 ). 把 它 与 归 纳 假 设 不 i ， 刁 飞 不 j 。 ） 组 合 ， 我 们 得 到 不 i 冖 ） < ， 00 ． 于 是 当 贫 心 算 法 
选 择 瞟 时 ， 此 刻 区 间 是 在 有 效 区 间 的 集 合 R 中 ． 心 算 法 试 择 且 南 最 小 结 束 时 画 的 有 效 区 
间 ； 因 为 是 这 些 有 效 区 间 中 的 一 个 ， 我 们 有 不 和 ） < 歹 00 ． 这 就 完 成 了 归 纳 步 骤 ． 
0 

**SUMMARIES**