LN4. 贪心算法. 区间调度问题.

* 贪心算法的思想：由多次迭代，多个决策步骤组成。

每次决策 { 短视(Myopic)，局部最优
　　　　　 不可逆(Irrevocable)：一旦做了决策，则不可改变。

* 贪心算法的一般流程：
  ① 对任务按某种优先级排序.　　$O(n\log n)$
  ② 按此排序逐个考查，做决策.　　$n\cdot O(1)$

* 贪心算法的挑战：算法正确性证明.（优先级的选择，很多优先级排序不一定最优）

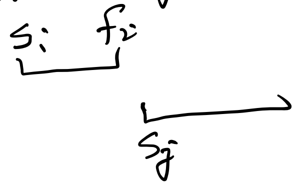
* 区间调度问题.

1. 问题定义：

已知：n 个区间 $I_1 \cdots I_n$. $I_j = [s_j, f_j]$, $s_j < f_j$.

目标：选择尽可能多的无冲突的区间.

Def. 无冲突的集合 S. $\forall i \neq j,\ i,j \in S.\ s_i \le s_j \Rightarrow f_i \le s_j$



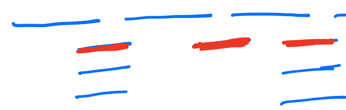
2. 应用：对资源的使用/利用.

例：

书 —— 借书人	车间机器 —— 完成任务	周耀杰
光碟 —— 借碟人	火车站站台 —— 列车进站停	李新毅
中心处理器 —— 任务请求	羽毛球场地 —— 同学预约	刘俊德
	外卖骑手 —— 外卖请求	陆墨涵

3. 如何贪心，即如何对任务排序：

① 区间短优先 反例: 

② 冲突少优先 反例:  丁树浩

③ 开始时间早优先 反例: 

④ 距离两端最近优先 反例: 同③

⑤ 结束时间早优先.

4. 贪心算法 (Earlist Finishing Time, EFT)

选中集合 $T \leftarrow \phi$

循环：∃可挑选的区间

- 选择其中结束时间最早的一个，加入调度到表 T
- 删除与之冲突的区间

返回 T

5. 算法的正确性证明：

t_r: 第 r 个调度区间的结束时间

C_r: 与第 r 个调度区间有冲突的区间集合.

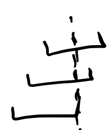
Claim1. C_r 中任一区间均包含 t_r.

$$\begin{cases} f_i \geq t_r \\ f_i < t_r \ \times \end{cases}$$

Claim2 任一区间，一定被某个 C_r 包含.

Thm. EFT 算法可得到最优解

Pf. $C_1, \cdots, C_m$
$t_1, \cdots, t_m$

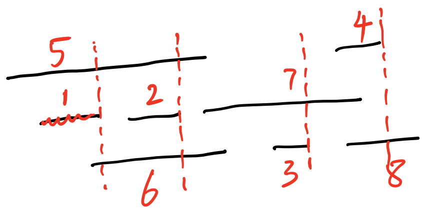
C_r: 

t_r

每个 C_r 集合中至少选出一个. ∵互相冲突

⇒ 最多选出 m 个不冲突的. [鸽笼原理]

例:



C_1 : 1, 5, 6
C_2 : 2
C_3 : 3, 7
C_4 : 4, 8

最多选4个

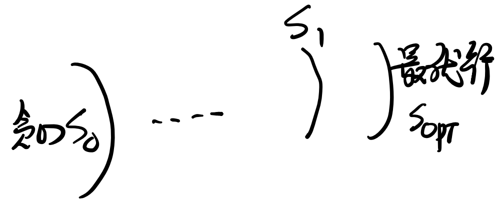
★ 贪心算法最优性证明方法: 交换论证.

① $\forall S$, $\exists S'$, S' 与 S_0 更近, $S' \geq S$

$\forall S$, $S \leq S' \leq S'' \leq \cdots \leq S_0$

② 对于 S_0, $\exists S_1$ 与 S_{OPT} 更近, $S_0 \geq S_1$

$S_0 \geq S_1 \geq S_2 \cdots \geq S_{OPT}$

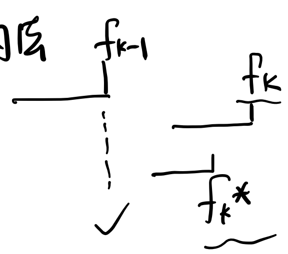


★ 用交换论证法证明EFT的正确性. 设 O^* 为 OPT

Claim. 根据第j个任务的结束时间, EFT总是"stays ahead" O^*

Lemma. 设 S 为 EFT 的调度结果. 则 $\forall j \in \{1, \cdots |S|\}$. $f_j \leq f_j^*$

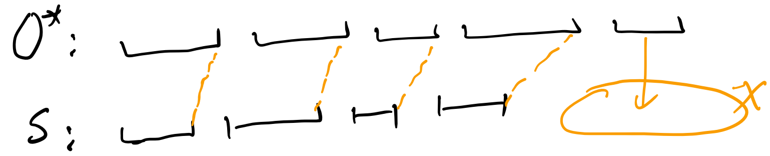
Pf: 归纳法



Lemma. 若 $\forall j \in \{1 \cdots k\}$, $f_j \leq f_j^*$. 则 S 是最优的

$k = |S|$.

Pf. 反证法:



L15. 带ddl的区间调度

* 定义:

已知: n个任务, 任务i: 长度 t_i, ddl: d_i, 启动时间 $s=0$.

求: $\forall i$, $[s_i, f_i]$, $f_i - s_i = t_i$, $s_i \geq s$

$\forall i \neq j$, 互不重叠

目标: $\min L = \min \max_i l_i$

$= \min \max_i (f_i \dot{-} d_i)$

$l_i = \begin{cases} f_i - d_i, & \text{if } f_i > d_i \\ 0, & \text{else} \end{cases}$

Def. $l_i = f_i \dot{-} d_i$

联想: $l_i = \mathrm{ReLU}(f_i - d_i)$

* 应用:

① (论文写作+实验研究) 会议投稿. $l_i = \begin{cases} \infty, & f_i > d_i \\ 0, & \text{else} \end{cases}$

② 作业的提交: 作业完成的预估长度, 提交截止时间

Q: 其它应用?

③ 期末复习考试: 复习时间, 在考试之前复习所有内容的学习. 王希宇

④ 占充电桩付费位: ddl: 充电完成时间, 对应延迟: 交钱 李明泓

⑤ 快递员送外卖: 在ddl前送到 周煜杰

⑥ 还贷款: n笔, 先还哪个, 逾期会增加利息. 肖家明

⑦ 建筑工程与工期: 项目, 工期, 超期有损失. 刘子墨

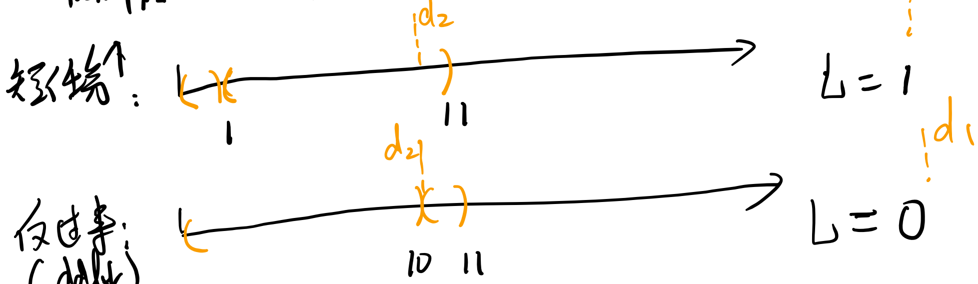
⑧ 飞机登机口与停机坪的上机（多人并行、不太适合）李新毅

⑨ 自然灾害、人员转移。周煜杰

* 安排贪心（贪心策略的设计）

1) 短任务优先：$T = \{(t_i, d_i) \mid i = 1 \dots n\}$

反例：(1, 100), (10, 10)

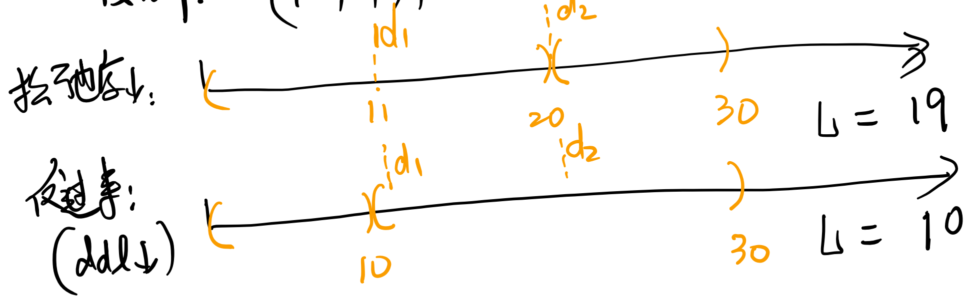


2) 紧迫度↑/松弛度↓优先：

Def. 松弛度 $:= d_i - t_i$

上例：2, 1　OK

反例：(10, 11), (20, 20)



3) ddl小优先，上两例都OK.

* 贪心算法

1: 按ddl从小到大排序.

2: 循环：依此顺序依次调度.

* 算法的正确性证明：

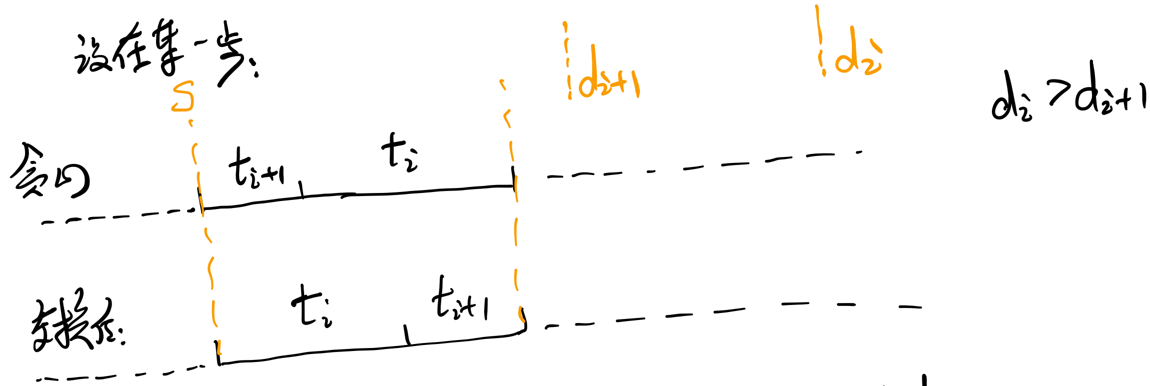
假设 ① 不失一般性，假设ddl均不同.

② 对于任何一个调度，可去掉任务间的空隙

设 贪心调度为 S_0，最优调度为 S_{OPT}: $1, 2, \cdots, n$

若 $S_0 \neq S_{OPT}$. $S_0 \to \cdots \to S_{OPT}$
冒泡排序，复杂度为 $O(n^2)$

设在某一步:



交换前: $f(i) = s + t_{i+1} + t_i$

$f(i+1) = s + t_{i+1}$

$l(i) = s + t_{i+1} + t_i \,\dot{-}\, d_i$

$l(i+1) = s + t_{i+1} \,\dot{-}\, d_{i+1}$

交换后: $f'(i) = s + t_i$

$f'(i+1) = s + t_i + t_{i+1}$

$l'(i) = s + t_i \,\dot{-}\, d_i$

$l'(i+1) = s + t_i + t_{i+1} \,\dot{-}\, d_{i+1}$

$\Rightarrow$ 已知 $d_i > d_{i+1}$:

$$\begin{cases} l'(i+1) \geq l(i) \\ l'(i+1) \geq l(i+1) \end{cases} \Rightarrow L' \geq L$$

$\Rightarrow$ 依此冒泡: $\underset{\text{贪心}}{S_0} \geq S_1 \geq S_2 \geq \cdots \geq \underset{\text{最优解}}{S_{OPT}}$