Greedy Algorithm 的策略

每个回合都選擇該回合的 local optimal choice

最後即可得到 global optimal choice

正確性來源：1. Optimal Substructure：原問題的 OPT 是由子問題的 OPT 所構成

利用 cut-and-paste 證明之

假設原問題的 OPT 不由子問題之 OPT 構成，則將該子問題解部份替換成

子問題之 OPT，則必可得比原問題 OPT 更優解，原問題 OPT 不為最佳解（→←）

2. Greedy-choice Property：原問題的其中一个最佳解中會包含有 greedy choice

利用 exchange argument 來證明之

設一 optimal solution 為 OPT，一 greedy choice 為 g

claim：OPT 中會包含 g，但共有2种 cases 如下：

1. g is in OPT，得證

2. g not in OPT：可以將 OPT 裡頭的一个 choice 替換成 g 得到 OPT'

證明 OPT' 至少會跟 OPT 一樣好

若一樣，則得證有存在一种 OPT 包含 g

若更好，則和原 OPT 為 optimal solution 矛盾

解題流程：1. 定義子問題，guess（窮舉所有子問題可能的解的情形）

2. 證明具有 optimal substructure

3. 選擇 greedy choice 的取法

4. 證明 greedy choice property

Interval Scheduling

問題定義：給定 n 个活动，n 个活动分別記作 a_i，則所成集合 $S=\{a_1,a_2,\dots,a_n\}$

已知各个 a_i 的開始時間為 s_i，结束時間為 f_i

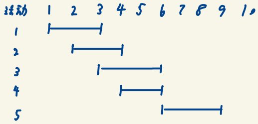
且滿足 $f_i \le f_j,\ \forall i<j$ 的關係

求這 n 个活动之最大可排程活动數為何？or 最大可執行集合為何？

Input：s_i 和 f_i $\forall a_i$ 且 $f_i \le f_j$ $\forall i<j$

Output：最大可排程集合 S_{max} 和 $|S_{max}|$

eg



則可知 $S_{ans} = \{a_1, a_3, a_5\}$ or $\{a_2, a_4, a_5\}$

$|S_{ans}| = 3$

① <u>定義子問題</u>: S_i 為考慮 $a_1 \sim a_i$ 時之最大可執行集合問題

⇒ 欲求 S_n，其中: S_i 之OPT應包含下面2種 case:

① OPT中有 a_i ⇒ 令滿足 $f_k \le s_i$ 之最大 k 為 j，則 OPT / $\{a_i\}$ 為 S_j 之OPT

② OPT中無 a_i ⇒ 則 OPT 會為 S_{i-1} 之 OPT ⇒ eg S_4 之OPT為 $\{a_1, a_3\}$，則 S_3 之OPT亦為 $\{a_1, a_3\}$

eg 以前面例子，S_5 之OPT 為 $\{a_2, a_4, a_5\}$
則: $\{a_2, a_4\}$ 會為 S_4 之OPT
S_3 之OPT為 $\{a_1, a_3\}$
則: $\{a_1\}$ 為 S_1 之OPT

② <u>證明Optimal Substructure</u>: ① 令 S_i 之OPT中包含有 a_i，而滿足 $f_k \le s_i$ 之最大 k 為 j

設 S_j 之OPT' 不為 OPT / $\{a_i\}$，則利用 OPT' ∪ $\{a_i\}$ 可得对 S_i 之更優解

OPT 不會為 S_i 之 optimal solution (—×—)

② 令 S_i 之OPT不包含 a_i，設 S_{i-1} 之 OPT' 不會為 OPT

則利用OPT' 可得对 S_i 之更佳解，OPT 不會為 S_i 之 Optimal Solution (—×—)

<u>遞迴關係式</u>: $r_i = \begin{cases} 0 & \text{if } i = 0 \\ \max(r_{p_i}+1, r_{i-1}) \end{cases}$ 其中 p_i 為 $1 \sim i$ 中可和 a_i 相容之最大 index (上面的 j)

③ <u>Greedy choice</u>: 每一次都選 case 1，即當下 S_i 的活動 a_i，會為 greedy choice

即: 選擇 a_i，令滿足 $f_k \le s_i$ 之最大 k 為 j，將子問題縮減為 S_j，且 S_j 之OPT' 為 OPT / $\{a_i\}$

<u>遞迴關係式</u>: $r_i = \begin{cases} 0 & \text{if } i = 0 \\ r_{p_i}+1 \end{cases}$

④ <u>證明 greedy choice property</u>: 1. 設对於 S_i 之 Optimal Solution 為 OPT，若 OPT中包含有選當前 a_i 的 greedy choice，則得證

2. 設OPT中不包含有選當前 a_i 的 greedy choice，i.e. OPT 為 S_{i-1} 之 OPT

令 S_{i-1} 之 Optimal Solution為 OPT'，可將 OPT' 中的 last activity 替換成 a_i，工作數目不變

一樣為 optimal solution OPT''，會為