Csillagászati megfigyelések

Adjunk mohó algoritmust, amely megoldja a feladatot, ha a megfigyelések tudományos súlya egyforma.

Algoritmus

- Igazolom a feladat páronként kompatibilis M-beli megfigyelések egy maximális elemszámú $\{m_{i_1}, m_{i_2}, \ldots, m_{i_k}\}$ részhalmazának meghatározása.
- T.f.h. befejezési idejük szerint monoton növekvően rendezettek a megfigyelések
  - ↳ ha nem, akkor rendezzük

$$f_1 \leq f_2 \leq \ldots \leq f_n$$

- A kiválasztott megfigyelések $\boxed{H}$ halmazába kerülnek
  - kezdetben $H := \emptyset$
- Használunk egy $\boxed{G}$ listát, mindig a még nem vizsgált megfigyeléseket tartalmazza, befejezési idő alapján mon. növ. rendezve
- Ismételjük a következőket, amíg G nem üres
  - vigyük át G-ből H-ba a legkisebb befejezési időpontú megfigyelést (MOHÓ)
  - töröljük G-ből az ezzel egyszerre kizáró megfigyelést, amíg kompatibilis megfigyelést nem találunk

Helyesség

- Legyen $\boxed{H} = \{m_{j_1}, m_{j_2}, \ldots, m_{j_l}\}$ az algoritmus által adott megoldás, $j_1 < j_2 < \ldots < j_l$
  - H-beli megfigyelések páronként kompatibilisek
- Belátjuk, hogy H egy optimális megoldása a feladatnak
- Legyen $\boxed{O} = \{m_{i_1}, m_{i_2}, \ldots, m_{i_q}\}$ a feladat egy optimális megoldása

- Megmutatjuk, hogy $f_{j_r} \le f_{i_r}$ $(1 \le r \le l)$

↳ teljes indukció

→ ha $r = 1$ ⟹ az algoritmus először a legkisebb befejezési idejű megfigyelést választja ✓

→ ha $r > 1$ és t.f.h. $f_{j_{r-1}} \le f_{i_{r-1}}$

⇒ O-beli megfigyelések páronként kompatibilisek, ezért $f_{i_{r-1}} \le s_{i_r}$ ⟹ $f_{j_{r-1}} \le s_{i_r}$

↓

ez azt jelenti, hogy m_{i_r} benne volt abban a halmazban, amiből az algoritmus a legkisebb befejezési időpontú m_{j_r} megfigyelést választotta ⟹ $f_{j_r} \le f_{i_r}$

– Indirekt t.f.h. H nem optimális megoldás, azaz $\ell < k$

- ekkor O tartalmaz egy $m_{\ell+1}$ megfigyelést is
- O-beli megfigyelések páronként kompatibilisek, ezért

$$f_{i_\ell} \le s_{i_{\ell+1}} \Rightarrow f_{j_\ell} \le s_{i_{\ell+1}}$$

→ ez azt jelenti, hogy, amikor G-ből töröltük, az m_{j_ℓ} megfigyeléssel együtt kizárt $m_{i_{\ell+1}}$ G-ben maradt ⇒ az algoritmus nem fejeződött be ↯

Költség

– kezdeti rendezés: $O(n \log n)$
– „Válogatás", törlés: $O(n)$

$\Big\}\ O(n \log n)$

Átlagos várakozási idő

Egy étterembe nyitáskor n vendég érkezik. A vendégek ismerik a választékot, ezért azonnal rendelnek. A $\boxed{V_i}$ vendég által rendelt étel elkészítése $\boxed{t_i}$ ideig tart. A szakács egyszerre csak 1 étel készítésével foglalkozik és ezt nem szakítja meg. Ha elkészült az étel, azonnal felszolgálják a megfelelő vendégnek.

Feladat: adjunk hatékony (mohó) algoritmust, annak eldöntésére, hogy a szakács milyen sorrendben készítse el az ételeket, ha azt szeretnénk, hogy $\boxed{w_i}$ várakozási idők átlaga minimális legyen.

Pl.: 3 vendég, 3 étel

$t_1 = 2$ a rendeléseket $(3, 1, 2)$ sorrendben

$t_2 = 4$ készítjük el $\Rightarrow$ várakozási idők

$t_3 = 3$

$$w_3 = 3$$

$$w_1 = w_3 + \overset{2}{t_1} = 5$$

$$w_2 = w_1 + t_2 = 9$$

átlag: $\dfrac{3 + 5 + 9}{3} = \dfrac{17}{3}$

Algoritmus

– A szakács elkészítési idő szerint monoton növekvő sorrendben készíti el az ételeket.

↳ Költség: $O(n \log n)$

Helyesség

– T.f.h. a rendelések elkészítési idejük szerint monoton növekvően rendezettek → ha nem, akkor rendezzük

$$t_1 \leq t_2 \leq \dots \leq t_n$$

– Az algoritmus a következőképp ütemezett

V_1 által rendelt étel
- kezdete $s_1 = 0$
- befejezése $f_1 = s_1 + t_1$

V_2 által rendelt
- kezdete $s_2 = f_1$
- befejezése $f_2 = s_2 + t_2$

⇒ belátjuk, hogy ez optimális

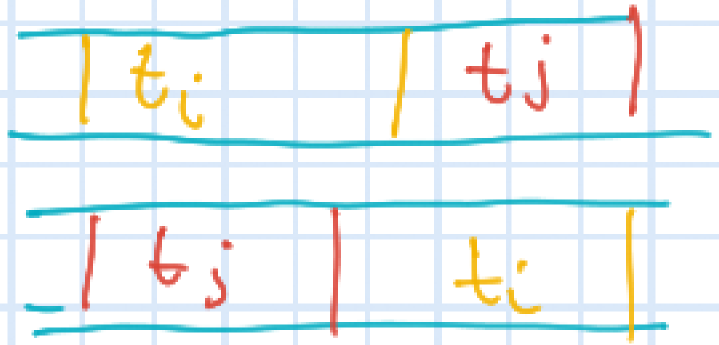
- Jelölje $\boxed{P}$ az algoritmus által előállított ütemezést
  - nincs olyan eset, hogy a szakács nem dolgozik, mikor még vannak elkészítendő ételek
- Az optimális megoldások között is kell lennie üresjárat nélkülinek, legyen $\boxed{O}$ egy ilyen
- Ha $O = P$ ✓
- Ha $O \neq P$
  - ekkor O-ban van két egymás után elkészített $\boxed{e_i}$ és $\boxed{e_j}$ étel, ahol $t_i \geq t_j$
  - cseréljük fel O-ban e_i és e_j ütemezését, legyen az így kapott ütemezés $\boxed{O'}$

- O'-ben V_i és V_j vendég várakozási ideje változik a többi vendégé nem

→ V_i étele akkor készül el, mint V_j étele a csere előtt

O:

O':



→ V_j étele a csere után mindenképp elkészül addig, mint a csere előtt V_i étele, hiszen $t_j \leq t_i$

⇒ így a várakozási idők összege (ekvivalens az átlaggal) O'-ben nem nagyobb, mint O-ban

→ Tehát az O optimális ütemezésből szomszédos ételek elkészítésének cseréjével eljuthatunk P-hez

- Mivel csere lépések során az átlagos várakozási idő sosem növekszik így P is optimális

↳ legfeljebb $\binom{n}{2}$ csere szükséges (buborék rendezés ötlete)