2021. november 30.

Ütemezés

Egy vállalkozót többen keresnek különböző munkákkal. Minden munkáról ismert az elvégzési ideje, a határidő, és a munkadíj, amit akkor kap meg a vállalkozó, ha a munkát határidőre elvégzi. Mely munkákat vállalja el a vállalkozó és azokat hogyan ütemezze, ha a bevételét maximalizálni akarja?

Matematikailag

$M = \{m_1, m_2, \dots, m_n\}$ munkák

Minden m_i munkához

t_i elvégzési idő

d_i határidő

w_i munkadíj

Az $m_{i_1}, m_{i_2}, \dots, m_{i_k}$ munkák egy megengedett ütemezése a munkákhoz olyan $[s_{i_1}, f_{i_1}]$, $[s_{i_2}, f_{i_2}], \dots, [s_{i_k}, f_{i_k}]$ intervallumok rendelése, hogy

$0 \le s_{i_1}$, $f_{i_1} = s_{i_1} + t_{i_1}$, $f_{i_1} \le d_{i_1}$

$f_{i_1} \le s_{i_2}$, $f_{i_2} = s_{i_2} + t_{i_2}$, $f_{i_2} \le d_{i_2}$

$f_{i_2} \le s_{i_3}$, $f_{i_3} = s_{i_3} + t_{i_3}$, $f_{i_3} \le d_{i_3}$

$\vdots$

Optimalitás

$w_{i_1} + w_{i_2} + \dots + w_{i_k}$ maximális

(Általában feltesszük, hogy az összes t_i és d_i pozitív egész, továbbá a w_i munkadíjak is pozitívak)

Ez egy NP-nehéz feladat; abban a spec. esetben, amikor az összes határidő ugyanaz, ez épp a hátizsák feladat.

Létezik (nem polinomiális) DP algoritmus a feladat megoldására.

Más spec. eset

Az összes munkadíj ugyanaz. Ekkor a lehető legtöbb munkát kell ütemezni. Erre van $O(n \log n)$ költségű mohó algoritmus!

Mohó algoritmus

Rendezzük először a munkákat a határidejük szerint monoton növekvően. Az általánosság megszorítása nélkül (átindexelés) feltehetjük, hogy

$d_1 \leq d_2 \leq \dots \leq d_n$

Egy H halmazba fogjuk gyűjteni a kiválasztott munkákat. Kezdetben $H = \emptyset$

H-hoz egymás után vesszük hozzá az m_i munkákat $(i = 1, 2, \dots, n)$

Ha így $\sum_{m_j \in H} t_j > d_i$ adódik, akkor töröljük

H-ból a leghosszabb elvégzési idejű munkát.

Végül a H-beli feladatokat ütemezzük a 0 időpillanattól, szünet nélkül, a határidők szerint növekvő sorrendben $\rightarrow$ $\mathcal{H}$

ütemezzük kanonikus módon

$\mathcal{H}$-ban minden munka befejeződik a határidejére vagy már korábban.

Állítjuk, hogy $\mathcal{H}$ optimális ütemezés (a munkák számára nézve)

Az állítást a munkák n száma szerinti teljes indukcióval végezzük

Ha $n=1$, akkor az állítás triviálisan igaz

Legyen ezután $n>1$, és tfh az állítás igaz $n-1$ munka esetén

Ha H az összes munkát tartalmazza, akkor triviálisan optimális.

Tegyük fel ezért, hogy H-ból hiányzik legalább egy munka.

Legyen m_k az a munka, amelyet először törölt a mohó algoritmus H-ból, mondjuk akkor, amikor m_i-t vettük hozzá H-hoz

<u>Állítás</u>

Van olyan σ kanonikus optimális ütemezés, hogy m_k ebben se szerepel.

<u>Bizonyítás</u>

Hamarosan

Hogyan használhatjuk az állítást H optimalitásának bizonyításához?

Nyilván $|\mathcal{H}| \leq |O|$

Tekintsük ezután azt a feladatot, amikor az $m_1, \dots, m_{k-1}, m_{k+1}, \dots, m_n$ munkákat kell ütemezni.

A mohó algoritmus erre is $\mathcal{H}$-t fogja adni!

Az indukciós feltevés szerint $\mathcal{H}$ optimális az $m_1, \dots, m_{k-1}, m_{k+1}, \dots, m_n$ munkákra.

O megengedett megoldás ugyanezekre (nincs benne az m_k ne felejtsük), így $|\mathcal{H}| \geq |O|$

Ennélfogva $|\mathcal{H}| = |O|$, vagyis $\mathcal{H}$ optimális megoldása az eredeti feladatnak.

Költség:

Kezdeti rendezés $O(n\log n)$ pl. kupacrendezéssel

H manipulációja: a H-beli munkákat egy kupacban tároljuk, kulcs az elvégzési idő. Ezen kívül tároljuk (egy segédváltozóban) a kupacban lévő munkák össz elvégzési idejét

m_i betétele a kupacba $\rightarrow O(\log n)$

segédváltozó := segédváltozó + t_i $\rightarrow O(1)$

segédváltozó $\overset{?}{>}$ d_i $\rightarrow O(1)$

m_k törlése a kupacból $\rightarrow O(\log n)$
(maxkupacból maxtöröl)

segédváltozó := segédváltozó $- t_k \rightarrow O(1)$

Ez n munkára szintén $O(n \log n)$

Jöjjön ezután az állítás bizonyítása

Legyen O egy optimális kanonikus ütemezés.

Ha O-ban nincs benne m_k, akkor ez jó is lesz.

Tfh m_k ott van O-ban.

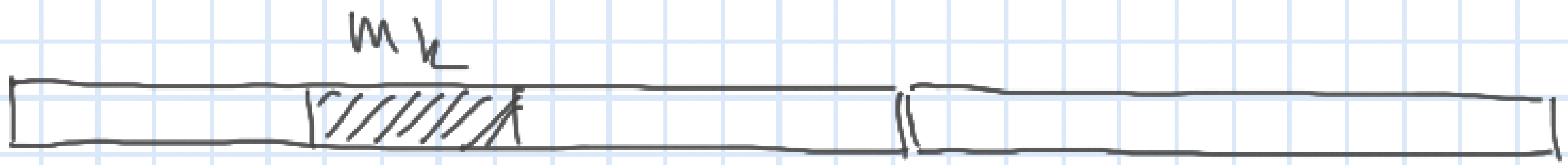
m_k definíciója szerint $\sum_{j=1}^{i} t_j > d_i$

Ezért az $m_1, \ldots, m_i$ munkák között kell olyannak lenni, amelyek nincsenek σ-ban, legyen m_ℓ egy ilyen, ahol $1 \le \ell \le i$.

Cseréljük le σ-ban m_k-t m_ℓ-re, és ütemezzük a munkákat kanonikusan $\longrightarrow \sigma^*$

Vegyük észre, hogy σ^*-ban is minden munka elkészül határidőre. Ez azon múlik, hogy $t_k \geq t_\ell$ (m_k egy leghosszabb elvégzési idejű munka volt $m_1, m_2, \dots, m_i$ közül).

σ :



$m_1, \dots, m_i$-ből ütemezettek $\quad$ $m_{i+1}, \dots, m_n$-ből ütemezettek

σ^* : m_k-t lecseréljük a nála rövidebb m_ℓ-re (ami nincs σ-ban); mindenki balra csúszik. így határidő sértés nem lesz továbbra se

Tételsor vizsgára (minden a Jamboardokról)

1. Kleinberg lehetetlenségi tétel
2. Gyors mátrixszorzás. Mester-tétel
3. Freivald algoritmus
4. Gyors polinomszorzás. FFT
5. DP vs. D&C
6. Optimális zárójelezés
7. Szekvenciaillesztés
8. Leghosszabb közös részsorozat

9. NP nehéz feladatok és DP
10. Hátizsák DP1
11. Hátizsák DP2
12. Hátizsák közelítő
13. Mohó ütemezés