Határidős feladatok ütemezése

Adottak feladatok egy $A = \{a_1, a_2, \dots, a_n\}$ halmaza, amelyek végrehajtása egy közös erőforrás használatával lehetséges.
Egyszerre csak 1 feladattal lehet foglalkozni és az nem szakítható meg. Minden a_i feladathoz adott t_i elvégzési ideje és egy d_i határideje.

Feladat: határozzuk meg A-beli feladatok egy maximális részhalmazát $\{a_{i_1}, a_{i_2}, \dots, a_{i_k}\}$ részhalmazát, amelyre a következők teljesülnek.

- minden a_{i_j} $(1 \le j \le k)$ feladathoz hozzárendelhető egy $s_{i_j} \ge 0$ kezdési és egy f_{i_j} befejezési időpont úgy, hogy $f_{i_j} - s_{i_j} = t_{i_j}$ és $f_{i_j} \le d_{i_j}$
- $[s_{i_j}, f_{i_j}]$ és $[s_{i_l}, f_{i_l}]$ intervallumok páronként diszjunktak $(1 \le j < l \le k)$

Algoritmus

- T.f.h. a feladatok határidejük szerint monoton növekvően rendezettek
  - ha nem, akkor rendezzük őket

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

- A kiválasztott feladatokat egy $\boxed{H}$ halmazban fogjuk tárolni
  - kezdetben $H := \emptyset$
- Minden $1 \leq i \leq n$ esetén
  - vegyük hozzá a_i-t H-hoz
  - ha $\sum_{a_j \in H} t_j > d_i$, akkor töröljünk egy leghosszabb elvégzési idejű feladatot H-ból
- H-beli feladatokat az első naptól kezdve, szünet nélkül, a határidejük szerint monoton növekvő sorrendben osszuk be

Helyesség

- Jelölje $|H|$ az algoritmus által adott ütemezést
  - bizonyítás, hogy H minden feladat határidőre elkészül
- A feladatok $|n|$ száma szerinti teljes indukcióval bizonyítjuk
  - helyességet (azt, hogy H optimális)
- Ha $n=1$ ✓
- Ha $n>1$, t.f.h. $n-1$ esetén H optimális
  - Ha H az összes feladatot tartalmazza, akkor kiindulásban optimális
  - t.f.h. van olyan feladat, amely nincs benne H-ban, legyen $|a_2|$, amelyet először hívott az algoritmus H-ból $|a_i|$ hozzávételekor

- t.f.h. $\boxed{O'_i}$ optimális megoldásban szerepel $\boxed{a_k}$, mivel $\sum_{j=1}^{i} t_j > d_i$ $a_1, a_2, \dots, a_i$ feladatok között van olyan $\boxed{a_l}$, amely nem szerepel O'-ben
- legyen $\boxed{O}$ az az ütemezés, amit úgy kapunk O'-ből, hogy az a_k feladatot kicseréljük a_l-re, majd a feladatokat az első naptól kezdve szünet nélkül ..... ütemezzük
- $a_1, a_2, \dots, a_i$ közül való O-beli feladatok a határidejükig elkészülnek, hiszen ugyanez teljesül, ha $a_1, \dots, a_{k-1}, a_{k+1}, \dots, a_i$-t osztjuk be, $t_k \geq t_l$ miatt $a_{i+1}, a_{i+2}, \dots, a_n$ közül O-beliek szintén elkészülnek időre
- ebből az következik, hogy O is optimális

- H-beli feladatok száma nem lehet több, mint O-beliek, mivel O optimális, azaz $|O| \geq |H|$
- Ha a mohó algoritmust $A \setminus \{a_k\}$ halmazra futtatjuk szintén H-t kapjuk

  $\Downarrow$

  az indukciós feltevés szerint ez optimális $A \setminus \{a_k\}$-ra
- O-ban is csak $A \setminus \{a_k\}$-beli feladatok vannak, így $|O| \leq |H|$

H-beli feladatok száma megegyezik O-beli feladatok számával

$\Downarrow$

H is optimális.

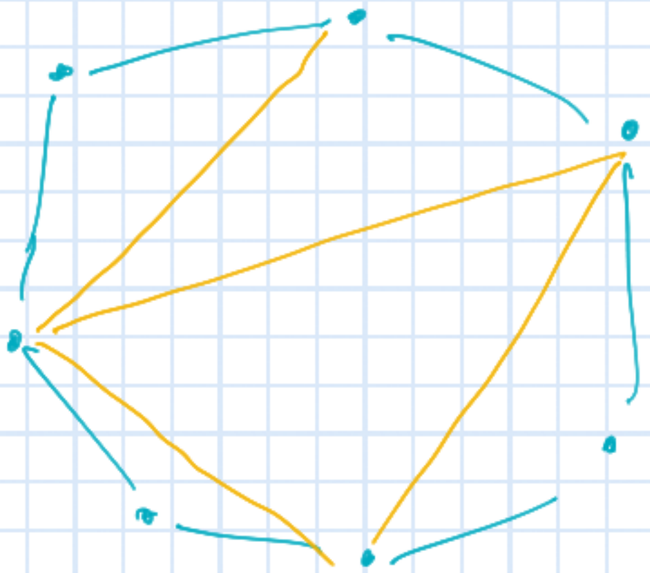
Költség

- H-beli feladatok nyilvántartása elvégzési idő szerint
  - célszerű maximum kupacot használni
    - → beszúrás, törlés: $O(\log n)$
    - → a maximum lekérdezés: $O(1)$
- Tartsuk nyilván H-ban lévő feladatok elvégzési idejének összegét
  - ⟹ törlés szükségessége $O(1)$ időben eldönthető
- Kezdeti rendezés: $O(n \log n)$
- összesítés: $O(n \log n)$

Optimális triangulációk

Egy n csúcsból álló konvex sokszöget szeretnénk háromszögekre bontani úgy, hogy a csúcsok közötti átlókat húzzuk. Az átlók nem metszhetik egymást.

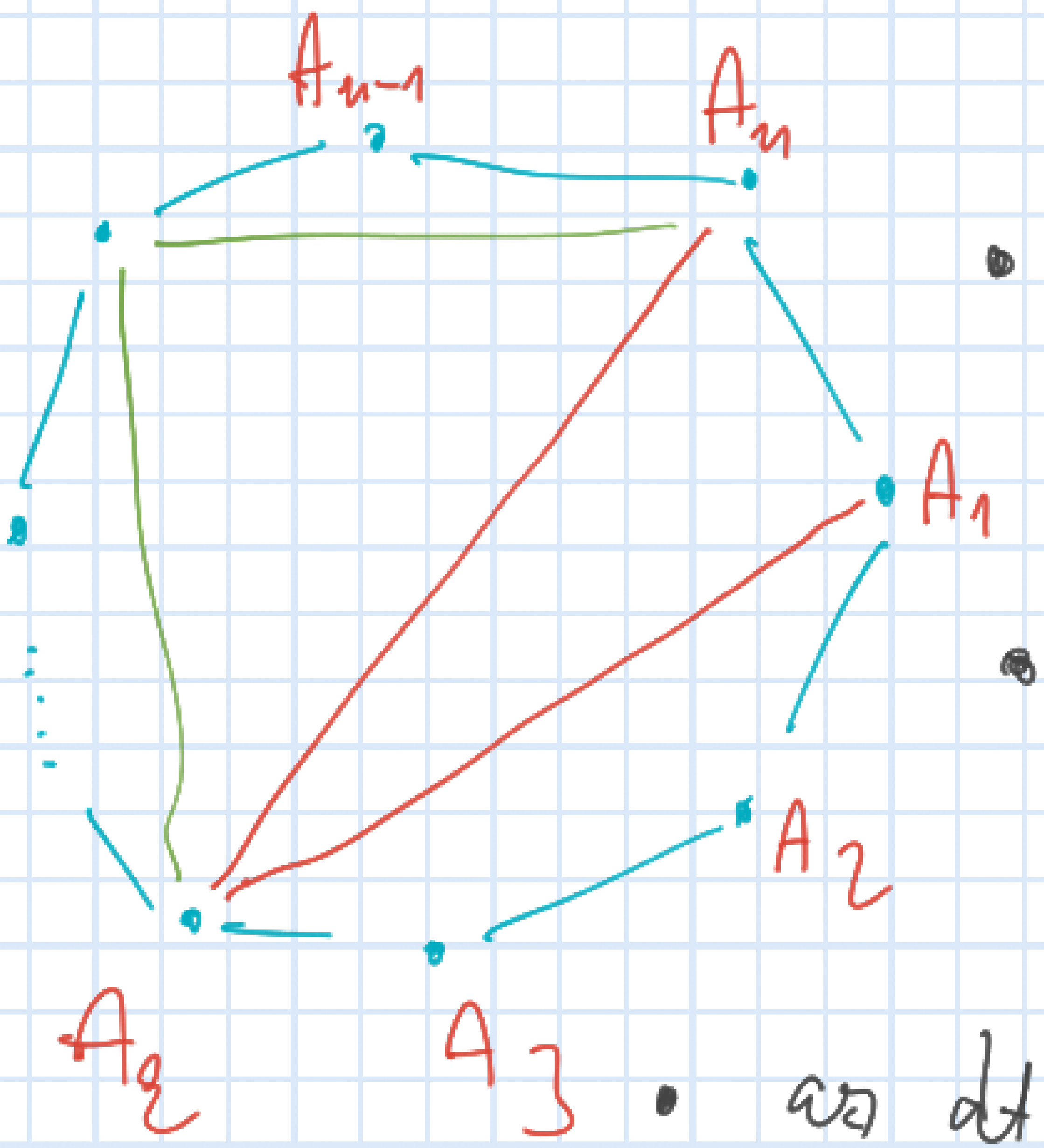
Feladat: a Δ-ek összterülete legyen minimális $\equiv$ az átlók összhossza legyen minim.



DP megoldás

— Részproblémák:

- konvex sokszöget egy egyenessel elvágva két konvex sokszöget kapunk



- tekintsünk egy olyan triangulációt, amelynek az egyik Δ-e $A_1 A_n$ oldalra illeszkedik.

→ legyen a 3. csúcs A_k ⇒ $A_1 A_n A_k$ Δ

- ezek után elég $\underbrace{A_1 A_2 \dots A_k}_{\text{konvex}}$ és $\underbrace{A_k \dots A_{n-1} A_n}_{\text{konvex}}$ sokszögek optimális triangulációját vizsgálni
- az átlók összhossza akkor minimális, ha a Δ-ek kerületének összege minimális
- legyen K[i,j] az $A_i A_{i+1} \dots A_j$ sokszög optim. triang. megtalálása

- Optimális részstruktúra
  - Tekintsük $R[i,j]$ feladat egy optimális megoldását, legyen az A_iA_j oldalra illeszkedő $\triangle A_iA_jA_k$
  - ekkor az optimális trianguláció $A_iA_{i+1}\cdots A_k$

    $A_k\cdots A_{j-1}A_j$ sokszögekre eső része optimális triangulációja ezen sokszögeknek.

    biz. . . . . .
- rekurzív m. o.
  - jelölje $R[i,j]$ optimális megoldásában az összkerület $C[i,j]$

$$C[i,j] = \begin{cases} 0, \text{ ha } i+2>j \\ \overline{A_iA_{i+1}} + \overline{A_iA_{i+2}} + \overline{A_{i+1}A_{i+2}}, \text{ ha } i+2=j \\ \min\limits_{k}\{C[i,k]+C[k,j]+(\overline{A_iA_j}+\overline{A_iA_k}+\overline{A_jA_k})\}, \text{ ha } i+2<j \end{cases}$$

$k \in [i+1 \ldots j-1]$

— a táblázatot dinamikusan kiírjuk ki, az optimális megoldás értéke C[1,n] cellában

- egy optimális megoldáshoz tároljuk el a választott i értéket

— Költség

- $O(n^2)$ cella
- egy cella kitöltése: $O(n)$
- összesen: $O(n^3)$