

COMPLEXITEIT 2025 HUISWERK 4

Deadline: Maandag 19 mei 23:59

Geef een duidelijke toelichting/uitleg bij al je antwoorden. Maak de uitwerking in L^AT_EX. Inleveren via Brightspace.

We bekijken de volgende twee beslissingsproblemen:

SUM: Gegeven een verzameling gehele getallen $S = \{s_1, \dots, s_n\}$ en een getal t . Bestaat er een $U \subseteq S$ waarvoor $\sum_{s \in U} s = t$?

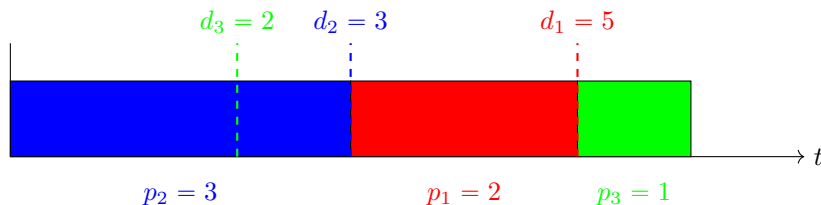
SCHEDULING1: Gegeven een verzameling taken $1, \dots, n$, die alle achter elkaar op één machine dienen te worden uitgevoerd, en een geheel getal B . Iedere taak heeft een verwerkingstijd p_i in minuten om de taak te verwerken. In totaal zijn er dus $\sum_{i=1}^n p_i$ minuten nodig om alle taken te verwerken.

Verder heeft iedere taak een deadline d_i en een gewicht w_i . Als taak i pas ná diens deadline wordt afgerond, dan betalen we een boete van w_i . Alle getallen dienen gehele getallen te zijn. Taken worden altijd volledig afgemaakt voor een volgende taak begint. Gevraagd is of er een ordening van de taken op de machine bestaat waarvoor de totale boete hoogstens B bedraagt.

Voorbeeld: Gegeven $B = 2$ en drie taken met de volgende verwerkingstijden, deadlines en gewichten:

taak	p_i	d_i	w_i
1	2	5	2
2	3	3	5
3	1	2	2

Dit is een ja-instantie van SCHEDULING1. Een volgorde waarin we de taken kunnen uitvoeren is 2, 1, 3. Zie de volgende afbeelding.



Taak 2 is dan na $p_2 = 3$ minuten klaar; dit is niet na de deadline $d_2 = 3$, dus geen boete. Taak 1 is na $3 + p_1 = 3 + 2 = 5$ minuten klaar. Gezien $d_1 = 5$ is dit ook op tijd. Ten slotte is taak 3 na $5 + p_3 = 5 + 1 = 6$ minuten klaar. Dit is te laat, want $6 > 2 = d_3$, dus we moeten een boete van $w_3 = 2$ betalen. In totaal betalen we dus een boete van 2. Dit is hoogstens gelijk aan B , dus de ordening voldoet.

a. (10 pt.) Toon aan dat SCHEDULING1 $\in \mathcal{NP}$ door een *niet-deterministisch polynomiaal* algoritme voor SCHEDULING1 te geven. Het algoritme heeft dus als invoer een rijtje verwerkingstijden $p = (p_1, \dots, p_n)$, een rijtje deadlines $d = (d_1, \dots, d_n)$, een rijtje boetes $w = (w_1, \dots, w_n)$ en een geheel getal B , en moet “ja” opleveren dan en slechts dan als de invoer (p, d, w, B) een ja-instantie is.

Leg onder andere uit wat in elke fase van het algoritme gebeurt en in het bijzonder wat in fase 2 wordt gecontroleerd en hoe, en hoeveel stappen dat kost. Leg ook uit waarom

jouw niet-deterministische algoritme “ja” oplevert dan en slechts dan als (p, d, w, B) een ja-instantie is en waarom het polynomiaal is.

We bekijken nu een transformatie T die instanties van SUM transformeert naar instanties van SCHEDULING1.

We definiëren $n = |S|$ taken. Taak i heeft als verwerkingstijd $p_i = s_i$, en ook dezelfde boete $w_i = s_i$. De deadline van alle taken wordt gelijk aan $\sum_{i=1}^n s_i - t$. Ten slotte kiezen we $B = t$.

Voorbeeld: Beschouw een instantie van SUM gegeven door $S = \{1, 2, 6\}$ en $t = 4$. Deze wordt door T omgezet in een instantie van SCHEDULING1 met geheel getal $B = 4$ en drie taken met de volgende kenmerken (merk op dat $\sum_{i=1}^3 s_i - t = 1 + 2 + 6 - 4 = 5$):

taak	p_i	d_i	w_i
1	1	5	1
2	2	5	2
3	6	5	6

b. (5 pt.) Laat zien dat de constructie van $T((S, t))$ polynomiaal begrensd is in $|(S, t)|$.

c. (10 pt.) Bewijs dat (S, t) een ja-instantie is voor SUM dan en slechts dan als $T((S, t))$ een ja-instantie is voor SCHEDULING1.

Uit opgaven b en c volgt dat $\text{SUM} \leq_P \text{SCHEDULING1}$. Uit college weten we dat $\text{SUM} \in \mathcal{NPC}$. Hieruit kunnen we dus concluderen dat $\text{SCHEDULING1} \in \mathcal{NPC}$.

d. (5 pt.) Kunnen we nu ook concluderen dat $\text{SCHEDULING1} \leq_P \text{SUM}$? Leg uit.

e. (5 pt.) Gegeven een willekeurig ander probleem $Q \in \mathcal{P}$. Is het wel, niet, of misschien mogelijk dat $\text{SCHEDULING1} \leq_P Q$? En $Q \leq_P \text{SCHEDULING1}$?

Ten slotte kijken we nog naar het volgende beslissingsprobleem.

SCHEDULING2: Gegeven een verzameling taken $1, \dots, n$, die alle achter elkaar op één machine te dienen worden uitgevoerd. Iedere taak heeft een verwerkingstijd p_i in minuten om de taak te verwerken.

Verder heeft iedere taak een deadline d_i en een vrijgavetijd r_i . Het is niet toegestaan om aan een taak te werken vóór diens vrijgavetijd of ná diens deadline. Taken worden altijd volledig afgemaakt voor een volgende taak begint. Gevraagd is of er een ordening van de taken op de machine bestaat die aan de gegeven beperkingen voldoet.

Voorbeeld: Gegeven als invoer drie taken met de volgende verwerkingstijden, deadlines en vrijgavetijden:

taak	p_i	d_i	r_i
1	3	7	2
2	2	11	8
3	4	5	0

Dit is een ja-instantie van SCHEDULING2. Een volgorde waarin we de taken kunnen uitvoeren is 3, 1, 2. We beginnen op $t = 0$ aan taak 3, die op $t = p_3 = 4$ klaar is, voor diens deadline $d_3 = 5$. Vervolgens mogen we beginnen aan taak 1 (want $r_1 = 2 \leq 4$), welke op $t = 4 + p_1 = 4 + 3 = 7$ klaar is, ook op tijd voor de deadline $p_1 = 7$. Nu staat de machine 1 minuut stil, waarna we op $t = 8 = r_2$ mogen beginnen aan taak 2. Deze is klaar op $t = 8 + p_2 = 8 + 2 = 10$, ook op tijd voor $d_2 = 11$.

f. (5 pt.) Bewijs dat $\text{SUM} \leq_P \text{SCHEDULING2}$. Doorloop zelf alle noodzakelijke stappen in voldoende detail.

Hint: Voeg één slim gekozen taak toe.