Complexiteit 2025
Huiswerk 4

Uitwerkingen

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 verwekingstijden $p = (p_1, \ldots, p_n)$, een rijtje deadlines $d = (d_1, \ldots, d_n)$, een rijtje boetes $w = (w_1, \ldots, w_n)$ 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.

Uitwerking:
We geven een algoritme in drie fases.

Fase 1: gokfase. We schrijven een willekeurige string s, verder te interpreteren als rij gehele getallen.

Fase 2: verificatiefase. We verifiëren of s een oplossing van de invoerinstantie representeert, d.w.z. een ordening van de taken die voldoet aan de gestelde eisen. Hiertoe controleren we de volgende zaken.

1. Staan er precies n getallen? Hiervoor lopen we hoogstens de hele string door, dus $O(|s|)$.
2. Is elk getal minstens 1 en hoogstens n? Wederom $O(|s|)$.
3. Komt ieder getal tussen 1 en n precies één keer voor? Een naïeve implementatie vergelijkt elk paar getallen in s, wat $O(|s|^2)$ stappen geeft; het kan sneller, maar in ieder geval polynomiaal in $|s|$.
4. Geeft de ordening van de taken in s een schedule waarin de totale boete hoogstens B is? Voor het gemak nemen we aan dat de invoer x bestaat uit drie arrays p, d en w en het gehele getal B (lijsten kan ook; hierdoor kosten een aantal vergelijkingen een factor $|x|$ meer stappen, maar dat maakt voor de polynomialiteit niet uit). We lopen één keer door s heen, en houden de totaal tot nu toe gespendeerde tijd bij (start met $t = 0$) en de totale tot nu toe opgebouwde boete (start met $b = 0$). Voor elke taak i tellen we de verwerkingstijd p_i op bij t, controleren we of de taak nu te laat is door t te vergelijken met d_i, en zo ja tellen we de boete w_i op bij b. Elk van deze operaties is lineair in de lengte van de invoer, dus $O(|x|)$. We doen deze operaties voor elke taak, dus in totaal $O(|x| \cdot |s|)$. Tenslotte kijken we of b hoogstens B is, ook weer $O(|x|)$.

Als elk van de vier tests doorstaan wordt, geven we "true" als uitvoer. Anders "false" of beginnen we een oneindige loop.

Fase 3: uitvoerfase. Als er in fase 2 "true" werd geretourneerd, antwoorden we "ja". Anders geen uitvoer.

Er geldt nu dat in de derde fase "ja" wordt geantwoord d.e.s.d.a. de gegenereerde string s een oplossing is voor het probleem, wat mogelijk is d.e.s.d.a. de invoer x een ja-instantie is voor het probleem.

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

Uitwerking:
Het opschrijven van de verwerkingstijden en boetes is tweemaal $O(|S|)$. Het berekenen van de deadline van alle taken kan in $O(|S| + |t|)$, en vervolgens schrijven we deze $|S|$ maal op, dus in totaal $O(|S| \cdot (|S| + |t|))$. Tenslotte kopiëren we t, wat $O(|t|)$ kost. Alles is dus polynomiaal begrensd 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.

Uitwerking:
Stel dat (S,t) een ja-instantie is voor SUM. Laat $I \subseteq \{1,\ldots,n\}$ een deelverzameling zijn waarvoor $\sum_{i\in I} s_i = t$. Maak nu een schema van de taken door eerst in willekeurige volgorde alle taken op de machine te zetten die níét in I zitten, en vervolgens in willekeurige volgorde de taken die wél in I zitten. Merk op dat

$$\sum_{i\notin I} p_i = \sum_{i\notin I} s_i = \sum_{i=1}^{n} s_i - \sum_{i\in I} s_i = \sum_{i=1}^{n} s_i - t = d_i,$$

dus de taken die niet in I zitten, zijn allemaal op tijd. Tevens zijn alle taken die daarna zijn gepland, dus precies de taken in I, allemaal te laat. Dit geeft een boete van $\sum_{i\in I} w_i = \sum_{i\in I} s_i = t = B$. Het beschreven schedule is dus een oplossing voor de geconstrueerde instantie $T((S,t))$, dus dit is een ja-instantie.

Stel nu dat $T((S,t))$ een ja-instantie is voor SCHEDULING1. Merk op dat de totale verwekingstijd van alle taken $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} s_i$ is, terwijl de deadline van alle taken gelijk is aan $\sum_{i=1}^{n} s_i - t$. Hoe de taken ook worden geordend, er wordt dus altijd een boete van minstens t ontvangen. Een boete van exact t is alleen mogelijk als de laatste taak die op tijd is precies op de deadline klaar is. Er moet dus kennelijk een verzameling taken zijn, zeg J, zodanig dat $\sum_{i\in J} p_i = \sum_{i=1}^{n} s_i - t$. Dan geldt ook dat

$$\sum_{i\notin J} p_i = \sum_{i=1}^{n} p_i - \sum_{i\in J} p_n = \sum_{i=1}^{n} s_i - \left(\sum_{i=1}^{n} s_i - t\right) = t.$$

De verzameling $\{s_i\}_{i\notin J}$ is dus een oplossing van de instantie (S,t) van SUM, dus deze instantie is een ja-instantie.

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

Uitwerking:
Ja, dit volgt uit het feit dat SUM $\in \mathcal{NPC}$ en het resultaat van opgave a dat ons vertelt dat SCHEDULING1 $\in \mathcal{NP}$.

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

Uitwerking:
Het is onwaarschijnlijk dat SCHEDULING1 $\leqslant_{\mathrm{P}} Q$ geldt, want dan zou Q zowel in $\mathcal{P}$ zitten als NP-volledig zijn, wat zou aantonen dat $\mathcal{P} = \mathcal{NP}$.

Het is wel waar dat $Q \leqslant_{\mathrm{P}}$ SCHEDULING1. Uit $Q \in \mathcal{P}$ volgt immers ook dat $Q \in \mathcal{NP}$, dus Q is reduceerbaar naar elk NP-volledig probleem, en in het bijzonder naar SCHEDULING1.

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

Uitwerking:
We maken een transformatie T' die instanties van SUM transformeert naar instanties van SCHEDULING2.

We definiëren eerst $n = |S|$ taken. Taak i heeft als verwerkingstijd $p_i = s_i$. Al deze taken hebben deadline $d_i = 1 + \sum_{i=1}^{n} s_i$ en vrijgavetijd $r_i = 0$. Vervolgens voegen we een extra taak $n+1$ toe met verwerkingstijd $p_{n+1} = 1$, deadline $d_{n+1} = t+1$ en vrijgavetijd $r_{n+1} = t$.

Merk allereerst op dat T' polynomiaal is in $|(S,t)|$. Voor de eerste n taken gaat het opschrijven van de verwerkingstijden in $O(|S|)$, het berekenen van de deadlines en deze vervolgens opschrijven in $O(|S| + |S|) = O(|S|)$, en het opschrijven van de vrijgavetijden in $O(|S|)$. Het toevoegen en opschrijven van de extra taak gaat in $O(|t|)$.

Stel nu dat (S,t) een ja-instantie is van SUM. Laat $I \subseteq \{1,\ldots,n\}$ een deelverzameling zijn waarvoor geldt dat $\sum_{i\in I} s_i = t$. Maak nu een schema waarin eerst de taken in I in willekeurige volgorde worden gescheduled, vervolgens taak $n+1$, en ten slotte de rest van de taken in willekeurige volgorde. Merk op dat de taken in I precies na $\sum_{i\in I} p_i = \sum_{i\in I} s_i = t = r_{n+1}$ klaar zijn, dus taak $n+1$ kan daarna meteen beginnen, en is na $p_{n+1} = 1$ minuten klaar op de deadline $d_{n+1} = t+1$. Vervolgens wordt de rest van de taken afgehandeld en is alles op tijd klaar voor de deadline van $1 + \sum_{i=1}^{n} s_i = 1 + \sum_{i=1}^{n} p_i = \sum_{i=1}^{n+1} p_i$.

Stel ten slotte dat $T((S,t))$ een ja-instantie is van SCHEDULING2. Alle taken kunnen alleen op tijd zijn als de machine nooit stilstaat, want de deadline $d_i = 1 + \sum_{i=1}^{n} s_i$ van de taken 1 tot en met n is precies gelijk aan de som van de verwerkingstijden van alle $n+1$ taken. Taak $n+1$ moet vanwege de vrijgave- en verwerkingstijd precies na t minuten worden gestart. Er bestaat dus een deelverzameling taken $J \subseteq \{1,\ldots,n\}$ waarvoor geldt dat $\sum_{i\in J} p_i = t$. Deze deelverzameling vormt dus ook een oplossing van (S,t), dus dit is een ja-instantie voor SUM.