

beCP 2019

Taak 1.2: Planning (scheduling)

Auteurs: Robin Jadoul, Damien Galant Voorbereiding: Damien Galant
Maximale uitvoeringsduur: 0s Geheugenlimiet: 0MB

Opmerking: Dit is een output only taak. Dit wil zeggen dat je enkel de resultaten van je berekeningen moet indienen, en niets van code. Er zijn verschillende input bestanden waarvoor je oplossing tesamen of apart kan indienen. Natuurlijk wil dit formaat ook zeggen dat je niet verplicht of ondersteund wordt hetzelfde programma voor elke input bestand te gebruiken. Daarnaast kan je misschien ook proberen of je sommige dingen met de hand kan oplossen.

De beCP trainers hebben recent veel werk gehad om deze wedstrijd voor te bereiden.

Er zijn T trainers. Er zijn nog N acties die uitgevoerd moeten worden voor alles klaar is (taken nalezen, testdata controleren, een paar bestanden naar de server uploaden, ...). De tijd dringt, dus willen ze al dit werk onderling verdelen zodanig dat de tijd wanneer alle trainers kunnen stoppen met werken en naar bed kunnen zo vroeg mogelijk is.

Jouw taak is om de trainers te helpen bij het onderverdelen van de acties die ze moeten uitvoeren. Er is geen specifieke volgorde tussen deze acties en elke trainer kan eender welke actie uitvoeren.

Input

De eerste lijn bevat 2 gehele getallen T, N : het aantal trainers en het aantal acties. De tweede lijn bevat N natuurlijke getallen: de duur d_i die elke actie nodig heeft om uitgevoerd te worden (in minuten).

Output

De eerste lijn van de output moet een enkel getal bevatten dat de tijd aangeeft wanneer de laatste trainer kan stoppen met werken. Dan moet je T lijnen output schrijven. De i de lijn begint met een geheel getal $k \geq 0$ dat het aantal acties aangeeft die de i de trainer moet uitvoeren. Dit wordt gevolgd door k getallen die de duur aangeven van de acties die trainer i moet uitvoeren.

Elke actie moet aan precies één trainer toegekend worden.

Algemene limieten

- $1 \leq T \leq 20$;
- $1 \leq N \leq 20000$;
- $1 \leq d_i \leq 100$;

Scores

Voor elk van de 10 tests, als je programma een tijd t geeft en t_{best} is de beste tijd in de test data, dan definiëren we de **ratio** van jouw oplossing als

$$r = \max\left(\min\left(\frac{t - t_{\text{best}}}{t_{\text{best}}}, 1\right), 0\right)$$

(Het kan zijn dat je betere oplossingen vindt dan die in de test data!)

De score die je krijgt is dan gegeven door de formule

$$\text{score} = 10^{1-10r}$$

In het bijzonder, als je oplossing minstens even goed is als die in de test data, dan krijg je 10 punten voor die test.

De totale score voor dit probleem is de som van de scores voor de 10 test, goed voor een maximum van 100 punten.

Voorbeeld 1

<p style="text-align: center; margin: 0;">sample1.in</p> <pre style="margin: 0;">2 5 1 2 3 4 5</pre>	<p style="text-align: center; margin: 0;">sample1.out</p> <pre style="margin: 0;">8 3 3 4 1 2 5 2</pre>
--	---

In dit voorbeeld zijn er 2 trainers en 5 acties. Een mogelijke optimale oplossing kent acties 1, 3, 4 toe aan de eerste trainer en taken 2, 5 aan de tweede. De eerste trainer stopt na $3 + 4 + 1 = 8$ minuten. De tweede trainer stopt na $5 + 2 = 7$ minuten. Het antwoord is dus 8 gezien alle trainers kunnen stoppen na 8 minuten.

Deze oplossing is niet uniek.

Merk op dat de taken toegekend aan elke trainer in willekeurige volgorde uitgeschreven kunnen worden.