

DAKI Programmeeropdracht 2: Bestelbus

(Deze versie is van 9 september 2022, 18:44)

Je moet deze programmeeropdracht **zelf** en **alleen** maken. Je mag zeker met anderen overleggen over je aanpak, maar code van anderen bekijken of overnemen of zelf code delen met anderen is uitdrukkelijk niet toegestaan. Je moet je programma schrijven in C#, en inleveren via [DOMjudge](#).

1 Opdrachtbeschrijving

Spullen bestellen op internet is tijdens de pandemie heel populair geworden, en is dat nog steeds. Nu er overal personeelstekorten zijn en meer chauffeurs onmogelijk te vinden zijn, moeten alle pakketjes van een internetwinkel worden vervoerd door één vrachtwagen met chauffeur, in een vast aantal ritten (in verband met de rijtijdenwet). Er kan dus geen vrachtwagen bijkomen want daar is geen chauffeur voor. Het is wél mogelijk om het laadvermogen van de vrachtwagen te veranderen. Je zet de DAKIBOT in om te bepalen wat het minimale gewicht is dat de vrachtwagen moeten kunnen vervoeren om een gegeven rij pakjes *in een gegeven volgorde* achtereenvolgens in een gegeven aantal ritten te kunnen vervoeren (zie het voorbeeld hieronder).

2 Invoer en uitvoer

De eerste regel van de invoer begint met twee positieve getallen: het aantal pakjes n (maximaal 99 999 999), en het aantal ritten r (maximaal 9 999 999). Na deze twee getallen kan nog wat staan, maar dat moet je negeren. Dan volgen één of meerdere regels met *samen* n getallen (elk $\leq 9\,999\,999$), het gewicht van een pakje. Gewichten zijn natuurlijk nooit negatief (hahahahaha, dat zou wat zijn!), maar omdat er vaak neutrino's worden besteld, moet je code wel goed met gewicht 0 kunnen omgaan.

De pakjes moeten in de gegeven volgorde worden vervoerd, in r ritten, met een zo licht mogelijke bestelbus. Schrijf code om de kleinst mogelijke (geheeltallige) capaciteit B te bepalen, zo dat je de rij van n getallen kunt verdelen in (hoogstens) r segmenten met som ten hoogste B . De uitvoer is het getal B , de verdeling in segmenten hoef je niet te geven.

3 Voorbeeld

Bij de volgende input:

```
10 5 (pakjes, ritten)
7 8 10 9 1 3 5
4 2 6
```

hoort de volgende output:

15

Als de volgorde niet van belang is, kan dit met een bus van omvang 11, want de paren $10 + 1$, $9 + 2$, $8 + 3$, $7 + 4$, en $6 + 5$ maken vijf ritten van gewicht 11. *Maar de volgorde is wél van belang*, en dan kun je rijden met $7 + 8$, 10 , 9 , $1 + 3 + 5 + 2 + 4$ en 6 (uiteraard ook vijf ritten), met een bus met laadvermogen 15. Met een bus van 14 gaat het niet (waarom niet eigenlijk?).

4 Algoritmische eisen

Je moet binair zoeken.

5 Hints, tips

Voor een zekere busomvang B kun je berekenen of die groot genoeg is: maak segmenten van hoogstens B en check of je r of minder van die segmenten nodig hebt. Met binary search kun je dan de kleinste B bepalen die genoeg is.