

# Checkpoints

Algoritmiek Opgave 2, Voorjaar 2024<sup>1</sup>

## 1 Achtergrond

Je hebt een game met  $n$  levels ontwikkeld, waarbij het  $t_i$  seconden kost om level  $i$  uit te spelen. Op de spelcomputer heb je ruimte om slechts  $K$  checkpoints op te slaan. Je wil weten na welke levels je een checkpoint moet plaatsen, om zo de tijd die een speler kan verliezen als hij dood gaat, te minimaliseren.

Een checkpoint kan alleen direct na het uitspelen van een level geplaatst worden, en als de speler een level verliest moet hij terug naar het begin van het eerste level na het laatst bereikte checkpoint.

## 2 Specificatie Invoer en Uitvoer

De invoer begint met een regel met drie getallen,  $n < 1.000.000$ , het aantal levels,  $K < 1000$ , het maximaal aantal te plaatsen checkpoints en  $m$ , de uitvoermodus (die 1 of 2 kan zijn). Daarna volgen een of meerdere regels met  $n$  getallen  $t_1, \dots, t_n$  - de tijd die het kost om level  $i$  uit te spelen. Er geldt dat  $t_i < 2^{31} - 1$ .

Indien de modus 1 is, is de uitvoer een regel met één getal: de maximale tijd die de speler zou kunnen verliezen, bij optimale plaatsing van de checkpoints, gevolgd door  $K + 1$  regels met ieder twee getallen. Het eerste getal op de  $i^e$  regel geeft het aantal levels voor het  $i^e$  checkpoint. Het tweede getal geeft de totale tijd die de speler kan verliezen als hij vlak voor het einde van het laatste level voor dit checkpoint dood gaat. Indien de modus 2 is is de uitvoer alleen de optimale waarde en hoeven (c.q. mogen) de plaatsing van de checkpoints en de lengtes van de segmenten niet gegeven te worden.

## 3 Voorbeeld Invoer en Uitvoer

Voorbeeld voorbeeld: Een klein voorbeeld waarbij het laatste segment het duurste is.

|                            |       |
|----------------------------|-------|
| 18 3 1                     | 276   |
| 88 68 50 80 13 71 68 38 71 | 3 206 |
| 2 52 30 41 15 75 93 32 76  | 5 270 |
|                            | 6 211 |
|                            | 4 276 |

Voorbeeld geen: Soms kun je geen checkpoints plaatsen.

|         |      |
|---------|------|
| 4 0 1   | 10   |
| 1 2 3 4 | 4 10 |

---

<sup>1</sup>Versie: 2 februari 2024.

**Voorbeeld Gelijk:** Hier zijn meerdere oplossingen mogelijk.

$$\begin{array}{r} 7\ 3\ 2 \\ 2\ 2\ 2\ 2 \\ 2\ 2\ 2 \end{array} \qquad 4$$

## 4 Algoritmische Aanwijzingen

Let er op dat je algoritme snel genoeg is, en bewijs (voor je gaat programmeren) dat jouw aanpak een voldoende snelle looptijd heeft. Een  $\Theta(n^2)$ -algoritme is niet goed genoeg, we zijn op zoek naar een algoritme met looptijd  $O(n \log(\sum_{i=1}^n t_i))$ .

Tip: probeer eerst een algoritme te bedenken dat in  $O(n)$  het volgende probleem oplost: gegeven  $T$ , bepaal of er een plaatsing van checkpoints mogelijk is waarbij je nooit meer dan  $T$  tijd verliest. Dat kun je misschien als subroutine gebruiken...