

Opdracht 2. Stagebezoeken

Een landelijk opleidingsinstituut heeft stagiaires op allerlei locaties. Op een speciale begeleidingsdag moeten deze allemaal worden bezocht. Er is een groep docenten die dat voor zijn rekening gaat nemen. Het instituut wil dit zo plannen dat er zo weinig mogelijk docenten nodig zijn en dat de reistijd wordt geminimaliseerd.

Van alle stageplaatsen is de locatie bekend in de vorm van een tweetal coördinaten. Datzelfde geldt ook voor de huisadressen van de docenten.

Voor de reistijd tussen plaatsen A met coördinaten (x_A, y_A) en B met coördinaten (x_B, y_B) geldt dat je die in deze opdracht kunt berekenen met de formule $|x_A - x_B| + |y_A - y_B|$. Deze tijd is in minuten.

Een stagebezoek duurt altijd precies 60 minuten.

Een docent mag van de instelling inclusief reistijd maximaal 500 minuten werken per dag.

Iedere docent kan iedere stageplaats bezoeken (de maximale reistijd van een willekeurige docent naar een willekeurige stageplaats is 220 minuten).

Invoer:

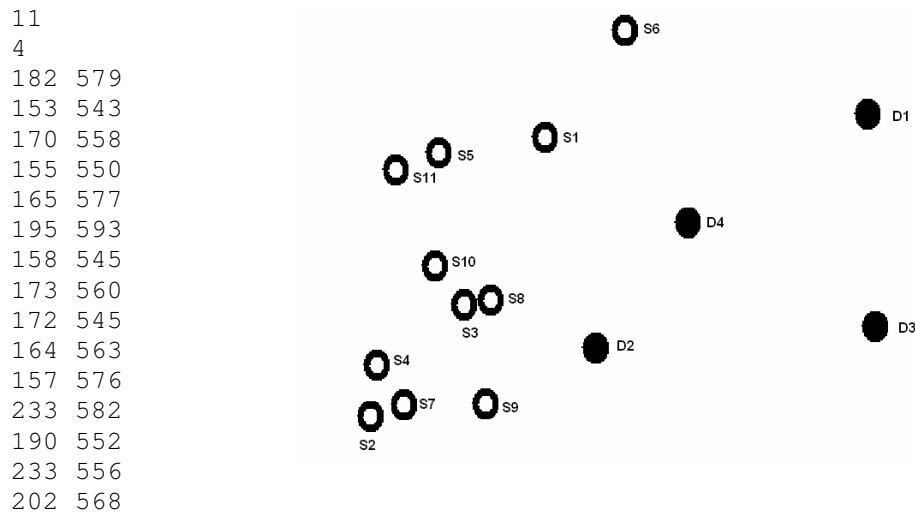
Invoer bij alle opdrachten is een bestand `stage.in` dat bestaat uit een groot aantal tekstregels.

Op de eerste regel staat het aantal stageplaatsen s , met $1 \leq s \leq 100$.

Op de tweede regel staat het aantal docenten d , met $1 \leq d \leq s$.

De volgende s regels geven de stagelocaties van de studenten, één locatie per regel. De coördinaten zijn positieve gehele getallen en worden gescheiden door een spatie. De eerste van deze regels hoort bij student nummer 1, de volgende bij student nummer 2, tot en met de regel met student nummer s . De volgende d regels geven de huisadressen van de begeleiders, één locatie per regel. De coördinaten zijn positieve gehele getallen en worden gescheiden door een spatie. De eerste van deze regels hoort bij docent nummer 1, de volgende bij docent nummer 2, tot en met de regel met docent nummer d .

Voorbeeld bestand `stage0.in`:



Dit bestand wordt bij de voorbeelden bij de opgaven als invoer gebruikt.

De afbeelding geeft een schetsmatige weergave van de Stageplaatsen (S met nummer) en de Docentadressen (D met nummer) uit het voorbeeld.

Voorbeeldbestanden en testen:

Er zijn bestanden `stage0.in`, `stage1.in` tot en met `stage5.in` beschikbaar waarmee je je programma kunt uitproberen.

Er is een batchfile `test2.bat`, die je kunt gebruiken op de volgende manier:

```
test2 nio2a stage0.in
```

Met deze opdracht test je het programma `nio2a` (of op deze plaats één van je andere programma's), waarbij vooraf eerst de invoer uit `stage0.in` (of op deze plaats één van de andere bestanden) naar het bestand `stage.in` wordt gekopieerd. Je zult dan zelf moeten controleren of het programma binnen de tijdlimiet stopt en de goede uitvoerfile maakt.

Opdracht 2. Taakoverzicht.

Opdracht	Programma	Invoer	Uitvoer	Tijdlimiet	Testen	Punten/test	Totaal
2A	<code>nio2a</code>	<code>stage.in</code>	<code>2a.uit</code>	1 s	6	3	18
2B	<code>nio2b</code>	<code>stage.in</code>	<code>2b.uit</code>	1 s	6	4	24
2C	<code>nio2c</code>	<code>stage.in</code>	<code>2c.uit</code>	1 s	6	5	30
2D	<code>nio2d</code>	<code>stage.in</code>	<code>2d.uit</code>	3 s	4	7	28

Opdracht 2A: Het verst verwijderde stageadres

Schrijf een programma `nio2a` dat een bestand `stage.in` inleest. Uitvoer is een bestand `2a.uit` van één regel. Hierop staat het nummer van het stageadres dat de grootste reistijd heeft voor de dichtst bij wonende docent. Als je kunt kiezen tussen verschillende stageadressen kies je het laagst genummerde!

Voorbeeld: 11

Opdracht 2B: Reistijd docenten naar dichtst bijgelegen stageadres

Schrijf een programma `nio2b` dat een bestand `stage.in` inleest. Uitvoer is een bestand `2b.uit` van d regels. Voor iedere docent wordt aangegeven hoeveel tijd hij kwijt is als hij alleen naar het stageadres het dichtst bij zijn huisadres gaat, daar een bezoek aflegt en meteen weer terugreist.

Voorbeeld: 158
110
188
122

Opdracht 2C: Uitbreiden met de minste kosten

Schrijf een programma `nio2c` dat een bestand `stage.in` inleest. Uitvoer is een bestand `2c.uit` van `s` regels. Op iedere regel staat het nummer van een stageplaats, gevolgd door een spatie en het nummer van de docent die op bezoek komt. De regels staan op de volgorde dat de bezoeken naar onderstaand voorschrift zijn ingepland. Als een student gezien de beperkingen geen bezoek kan krijgen staat er een 0.

Zoek bij de stageplaats die het verst weg ligt de dichtst bij wonende docent. Deze worden aan elkaar gekoppeld.

Zoek zolang dat binnen de tijd kan de dichtstbijzijnde stageplaats en voeg die achter aan de route voor deze docent toe. Houd er rekening mee dat bij in maximaal 500 minuten weer op zijn huisadres terug moet zijn!

Als de eerste docent is volgepland begin je met de volgende. Je volgt precies dezelfde werkwijze: Eerst de verste stageplaats bezoeken, dan telkens plaatsen toevoegen.

Als alle stageplaatsen bezocht zijn kun je stoppen. Als alle docenten vol zitten moet je stoppen, de overige studenten krijgen nu een 0 als volgnummer van de docent.

Als je kunt kiezen tussen verschillende stageadressen met gelijke afstand tot de plaats waar je bent met de reis, kies je altijd de laagstgenummerde!

Voorbeeld:

11	4
5	4
10	4
3	4
8	4
2	2
7	2
4	2
9	2
1	2
6	1

Zoals je ziet staan in dit bestand eerst alle bezoeken die door docent 4 worden afgelegd op de volgorde waarin ze zijn gepland, dan de bezoeken van docent 2 en tenslotte het enige bezoek van docent 1. Docent 3 kan deze dag iets anders doen.

De totale tijd voor de docenten is 1040 minuten, namelijk $148 + 462 + 0 + 430$.

Toelichting op de uitvoer:

Bij onderdeel A heb je al gevonden dat de stageplaats het verst van de dichtst bij wonende docent stageplaats 11 is (ten opzichte van docent 4). Docent 4 gaat daarom eerst naar stageplaats 11, daarna gaat hij telkens naar de dichtst bijzijnde stageplaats, tot hij als hij die nog zou bezoeken in totaal meer dan 500 minuten werkt en reist. Hij gaat dus naar 5, 10, 3 en 8. De volgende kandidaat zou 9 zijn, maar dan is zijn totale tijd (hij moet ook nog weer naar huis) meer dan 500. Dus hier stopt zijn programma.
Nu doe je hetzelfde met docent 2, die achtereenvolgens naar 2, 7, 4, 9 en 1 kan.
Stageplaats 6 is nog over en die wordt bezocht door docent 1.

Opdracht 2D: Kan het nog beter?

Schrijf een programma `nio2d` dat een bestand `stage.in` inleest. Uitvoer is een bestand `2d.uit` met dezelfde structuur als de uitvoer bij 2C. De gevonden oplossing moet beter zijn dan die van het algoritme van opdracht 2C.

Een oplossing is beter dan een andere wanneer:

- Er meer stageplaatsen kunnen worden bezocht
- Of bij gelijk aantal stagebezoeken een kleinere totale reis- en werktijd voor de docenten heeft.

Bij alle testgevallen is dat mogelijk.

Als je de beste oplossing die de jury heeft gevonden weet te herhalen krijg je 7 punten voor een testgeval. Kom je nog goedkoper uit, dan kun je bonuspunten krijgen. Ook een gedeeltelijke score voor een resultaat dat beter is dan bij 2C maar minder dan dat van de jury is mogelijk.

Voorbeeld:

10	4
11	4
5	4
1	4
6	4
9	2
7	2
2	2
4	2
10	2
3	2
8	2

Bij 2C waren alle stageadressen al bezocht.

Bij deze voorbeeldoplossing is de totale tijd voor docenten is nu 928, namelijk $0 + 478 + 0 + 450$. Daarom is deze oplossing beter dan die bij 2C.