Belgische Olympiades in de Informatica

Vraag 1 – Opwarming

Q1(a) n

Een manier om dit probleem op te lossen is om op papier het geval $n = 3$ uit te werken. Een andere manier was om erachter te komen dat deze berekening de som maakt van alle elementen in een vierkante $n \times n$ matrix, waarvan de de elementen boven de hoofddiagonaal -1 waren en de elementen op of onder de diagonaal $+1$. Men ziet snel dat in zo'n matrix alle elementen buiten de hoofddiagonaal elkaar opheffen. Er blijft alleen de diagonaal over, die n elementen telt.

Q1(b) 31

Hier moest je er rekening mee houden dat de lus nog eenmaal doorliep als i exact gelijk werd aan 15. Je mag je in de berekening zelf uiteraard ook niet vergissen.

Q1(c) not (a % 2 = 0 or b % 2 = 0)

Het modulo-symbool (%) werd uitgelegd op de eerste bladzijde. "a is oneven" kunnen we schrijven als "a % 2 = 1" of ook "not (a % 2 = 0)". De gegeven uitdrukking kon je dan uitdrukken als "not (a % 2 = 0) and not (b % 2 = 0)" wat equivalent is met "not (a % 2 = 0 or b % 2 = 0)".

Vraag 2 – Kaartspel

Q2(a) secundair 3 4 1 **Q2(a) hoger** 3 5 2

Q2(b) secundair 2 3 4 **Q2(b) hoger** 2 4 5

Q2(c) secundair 4 3 2 **Q2(c) hoger** 4 3 1

Om dit probleem op te lossen, is het voldoende dat je op C steeds het grootste cijfer legt van alle kaarten op A en de eerste (meest linkse) op B. Als de grootste kaart op tafel A ligt maar niet op de eerste plaats, moet je operatie Y uitvoeren totdat dit het geval is. Zodra dit cijfer in de eerste positie staat op A of B, verplaats je het naar C. Blijf dit systeem aanhouden totdat alle kaarten op C liggen.

Vraag 3 – Leve de wind . . .

Deze vraag bevatte per ongeluk een fout. Het voorgestelde algoritme moest b teruggeven en niet a, om te kloppen voor alle mogelijke scenario's. Om niemand te straffen die omwille van deze fout in de opgave de oplossing niet zou hebben gevonden, werd deze vraag niet meegeteld voor kandidaten waarbij hun antwoord hun gemiddelde score zou hebben verminderd. We zijn ook soepeler geweest bij het verbeteren van deze vraag, omwille van dezelfde reden. De oplossingen hieronder zijn correct als het algoritme b teruggeeft.

Q3(a) a != b

Q3(b) (a + b) / 2

De oplossing was een binaire zoektocht: neem als nieuwe positie steeds de het midden van de twee vorige posities. Op die manier kunnen we het aantal mogelijke waarden bij elke iteratie delen door 2.

Vraag 4 – Ceci n'est pas un sudoku (Dit is geen sudoku)

Q4(a) secundair 7 (of 6) **Q4(a) hoger** 4

Q4(b) secundair 7 **Q4(b) hoger** 10

Q4(c) secundair 5 **Q4(c) hoger** 12

Q4(d) secundair 5 **Q4(d) hoger** 10

Door een onduidelijkheid in de opgave was het mogelijk de vraag op verschillende manieren te interpreteren: ofwel neem je de waarde van het linkerbovenvak als startwaarde, ofwel start je altijd vanaf waarde 0 ongeacht wat er in dat vak staat. We hebben beide interpretaties geaccepteerd, daarom zijn er twee mogelijke antwoorden voor vraag Q4(a) voor het secundair.

Er zijn meerdere strategieën om dit probleem op te lossen. Bij een ervan beginnen we met het uitrekenen van de score in elk van de vakjes van de eerste rij en kolom. Dan berekenen we de score voor elk vakje waarvan de vakjes links ervan en erboven al berekend zijn, en we nemen de kleinste van de twee scores om verder te gaan. Deze strategie is efficiënt want we hoeven niet voor elk mogelijk pad telkens opnieuw de score uit te rekenen.

Vraag 5 – Waar organiseren we de finale ?

Q5(a) Namen

Q5(b) secundair Bergen **Q5(b) hoger** Antwerpen

Q5(c) s + distance[i][j]

Q5(d) s / (N - 1)

Q5(e) ranking[k][1] > m

De hoofdgedachte achter het programma is dat we voor elk centrum:
- de som (variabele s) berekenen van de alle afstanden tot andere centra,
- het gemiddelde bereken, door te delen door het aantal andere centra
- de plaats in de gesorteerde lijst terugvinden waar we dit element moeten toevoegen, door k te verhogen totdat het element dat zich op k bevindt een grote gemiddelde heeft dan het te plaatsen element.

Vraag 6 – Het spel met de krijtjes

Q6(a) (remaining - 1)

of anders: **Q6(b)** remaining = 30 (of last = 0) **Q6(c)** 1 **Q6(d)** 4 - last

Dit is een variant op het spel Nim. Om te winnen, moet je ervoor zorgen dat je tegenstander achterblijft met 1 krijtje, of 5 krijtjes (want dan laat hij er voor jou 2, 3 of 4 over wat je toelaat om er terug 1 achter te laten voor hem), of 9 krijtjes (want dan laat hij er voor jou 6, 7 of 8 over wat je toelaat om er terug 5 achter te laten voor hem), of 13 krijtjes, 17, 21, 25, 29. Om zeker te zijn om te winnen, moet je tegenstander dus steeds $1 + 4x$ (x positief geheel getal) krijtjes hebben. Gezien we je gezegd hebben dat de beginnende speler gegarandeerd wint (we vertrekken van een startsituatie waarbij die garantie er is), is "(remaining - 1) % 4" de oplossing, en zal die nooit 0 worden.

Oplossing (a) werd correct gevonden door 17 van de 252 deelnemers, en (b-c-d) werd gevonden door nog eens 20 andere deelnemers.

Vraag 7 – Spreekt u *Zwendeltaal* ?

Q7(a)

```
for (i ← 0 to (end − start + 1)/2 − 1 step 1)
{
    tmp ← phrase[start + i]
    phrase [start + i] ← phrase [end − i]
    phrase [end − i] ← tmp
}
```

of anders:

```
for (i ← start to (end + start − 1)/2 step 1)
{
    tmp ← phrase [i]
    phrase [i] ← phrase [end + start − i]
    phrase [end + start − i] ← tmp
}
```

Bemerk dat als het om te keren woord een oneven aantal letters heeft, het niet nodig is om de middelste letter met zichzelf van plaats te wisselen.

Q7(b) phrase[i] = ' ' or phrase[i] = '.' (of anders: phrase[i] = ' ' or i = n - 1)

Q7(c) i - 1

Q7(d) wordstart ← i + 1

Vraag 8 – Numerieke Anagrammen

Oplossing met de gegeven sortering, zonder bonus:

```
sort(a,n)
sort(b,n)

for (i ← 0 to n−1 step 1)
{
    if (a[i] ≠ b[i])
    {
        return false
    }
}
return true
```

Oplossing om een bonus te verdienen:

```
count ← newarray (10)

for (i ← 0 to n−1 step 1)
{
    count[a[i]] ← count[a[i]]+1
    count[b[i]] ← count[b[i]]−1
}

for (i ← 0 to 9 step 1)
{
    if (count[i] ≠ 0)
    {
        return false
    }
}
return true
```

Er waren ook talloze andere oplossingen mogelijk.

Vraag 9 – Mijn planning voorbereiden (alleen hoger onderwijs)

Q9(a) $e - s$

Q9(b) $eerst = -1$ or $e \leq start$

Q9(c) $i \neq -1$

Q9(d) $volgende[plaats]$

Q9(e) $start\_i - e$