Werkcollege Programmeermethoden NA 14 november 2017

Gebruik ____ om één niveau van inspringen aan te duiden. Twee keer dit karakter betekent dus twee keer inspringen. Denk aan de dubbele punten!

1. *(aangepast van opgave 1 van het C++-tentamen van 9 januari 2006)*
We hebben een lijst `B` met verschillende float's, zoals: `2.0 0.6 3.1 7.2 6.2 6.5`.

a. Schrijf een Python-functie `gk`, aan te roepen als `gr, kl = gk(B)`, die in `gr` en `kl` de *lijst-indices* van het grootste en kleinste getal van `B` (met `len(B)` ≥ 2 elementen) oplevert. Er moet precies één for-loop gebruikt worden; gebruik geen `min()` of `max()`. (In het voorbeeld: 3 en 1.)

b. Schrijf een Python-functie `aantal = stijgdaal(B)` die bepaalt hoeveel *locale extremen* `B` heeft: lijst-elementen waarvoor beide directe buren kleiner zijn, of juist beide groter. Eerste en laatste element zijn per definitie ook locale extremen. (Voorbeeld: 5; `3.1` niet.)

c. Schrijf een Python-functie `opso(B)` die een `bool` oplevert die aangeeft of `B` oplopend gesorteerd is. De lijst mag alleen met de functies van **a** en **b** benaderd worden. (Voorbeeld: `False`.)

d. Schrijf een Python-functie `lengte = lang(B)` die bepaalt wat de lengte is van de langste stijgende serie aaneengesloten lijst-elementen. (Voorbeeld: 3, namelijk `0.6 3.1 7.2`.)

e. Als **c**, maar gebruik nu alleen de functie van **d**.

2. *(aangepast van opgave 3 van het C++-tentamen van 4 januari 2013)*
Gegeven is een `m` bij `m` (`m` > 0) Booleaans NumPy array `T`. Hierbij geeft `T[i,j]` aan of er een rechtstreekse trein van station `i` naar station `j` rijdt (`True`) of niet (`False`). Er gaat nooit een directe trein van een station naar zichzelf. De variabele `m` mag in de te schrijven functies worden gebruikt alsof het als globale ("constante") variabele is gedefinieerd.

`False`	`False`	`True`
`True`	`False`	`True`
`False`	`True`	`False`

a. Schrijf een Python-functie `duos(T)` die berekent hoeveel duos (`i`, `j`) met $0 \leq$ `i` $<$ `j` $<$ `m` er in `T` zijn, waarvoor geldt dat er zowel een rechtstreekse trein van `i` naar `j` is als van `j` naar `i`. In het voorbeeld: 1, namelijk $(1, 2)$.

b. Schrijf een Python-functie `druk(T)` die het nummer (als `int`) van het station met de meeste in- en uitgaande directe verbindingen geeft. Als er meer stations met deze eigenschap zijn, geef dan dat met het hoogste nummer. In het voorbeeld: station 2 (3 directe verbindingen, net als 1).

c. Schrijf een Booleaanse Python-functie `bereik(T,i,j)` die precies dan `True` teruggeeft als je `j` vanuit `i` kunt bereiken met één overstap. Of er ook nog een rechtstreekse verbinding is, doet er niet toe. Neem aan dat $0 \leq$ `i`, `j` $<$ `m` en `i` $\neq$ `j`.

d. Schrijf een Python-functie `aantal(T,i)` die bepaalt hoeveel stations je, beginnende in station `i`, als volgt bezoekt, totdat je niet meer verder kunt. Je gaat steeds, met precies één overstap (gebruik de functie van **c**) naar het eerstvolgende station met een hoger nummer. In het voorbeeld: vanuit `i` $= 1$ reis je (met overstap in 0) naar 2, en klaar, met 2 bezochte stations. De stations waar je overstapt worden niet meegeteld.