Bvp Herexamen 2019

een sad bvp buizer

Augustus 2019

1 Theorie

- Wat is stapsgewijze verfijning + geef een voorbeeld.
- Gegeven is de volgende functie:
  def functie(a,b):
  rest = a
  quotient = 0
  while rest > b:
  indent maar ik ben brak in LaTeX rest − = b
  indent en ik ben moe quotient + = 1
  return quotient, rest

  Wat doet deze functie en bewijs de correctheid.
- Gegeven een algoritme met mergelist en mergesort, wat is de tijdscomplexiteit van mergelist. (multiple choice, keuzes waren O(1), O(n), O(n^2), O($log(n)$), O($n \cdot log(n)$)
- (niet helemaal zeker). er is een functie die de min/max waarden wilt berekenen maar verwisselte telkens de eerste en de laatste waarden, is dit een syntax fout? (de functie was niet gegeven, er was alleen maar een vage uitleg erover dus ik weet ook niet of dit helemaal is wat het moet zijn sorry :3)
- er waren nog 2 vragen maar ik weet niet meer wat dat die waren, maar je gaat dat zowiezo kunnen!

2 Pycharm Examen

Er is een sociale netwerk dat van bepaalde personen namen moet onthouden, en met wie zij allemaal bevriend zijn. Je moet een gepaste data-structuur bedenken om al deze informatie in de python-programma bij te houden, implimenteer ook de volgende functies die hieronder zijn opgelijst.

Bemerk:

- Je mag ervan uitgaan dat elke persoon een unieke naam heeft.
- Je hoef binnen de functies geen validatie van de parameters te doen. Je mag er bijvoorbeeld steeds vanuit gaan dat een string wordt gegeven als er een string gevraagd is.
- LET OP! Lees aandachtig de beschrijving van de functies! Als een verzameling (set) als return-waarde gevraagd wordt, is een lijst niet goed!
- zorg dat de .py-file die je indient geen compilatie-fouten bevat!

Gevraagde Functies
Het kan zijn dat er een paar details missen maar dit is ongeverhet examen en elke functie dat je moest kunnen, ook werd er een test file gegeven.
########
Twee functies voor het aanmaken en invullen van de data-structuur
########

def creeer_sn():
deze functie creert een lege data-structuur, d.w.z. een initiële data-structuur, zonder gegevens over personen en vriendschappen.

def voeg_persoon_toe (sn, naam_persoon):
deze functie voegt mensen toe in het sociale netwerk, maar zegt nog niks over zijn vrienden. De functie geeft de data-structuur terug.

########
Drie functies voor eenvoudige vragen aan de data-structuur.
########

def voeg_vrienden_toe (sn, naam_persoon1, naam_persoon2):
deze functie maakt een vriendschap tussen twee mensen. Er geldt ook dat als persoon 1 met persoon 2 bevriend is, het ook wederzijds is. Als een van de personen niet in de sociale netwerk is, geeft het de data-structuur onveranderd terug. Deze functie geeft de vernieuwde data-structuur terug.

def zit_persoon_in_sn(sn, naam_persoon):
deze functie kijkt of naam_persoon in de sociale netwerk zit. Het geeft een Boolean als return waarde.

def is_bevriend_met(sn, naam_persoon1, naam_persoon2):
deze functie kijkt of 2 personen rechtstreeks bevriend zijn met elkaar en geeft een Boolean als return waarde.

########
Enkele meer geavanceerde functies die de data-structuur ondervragen.
########

def is_bijna_bevriend_met(sn, naam_persoon1, naam_persoon2):
deze functie gaat na of persoon 1 bijna bevriend is met persoon 2. Dit wil zeggen dat ze elk 1 gemeenschappelijke vriend hebben. Als ze rechtstreeks bevriend zijn geeft deze functie False terug. Het geeft een Boolean terug als return waarde. Als een van de personen niet in de sociale netwerk zit, geeft het ook False terug.

def populairste_lijst(sn):
deze functie bekijkt de data-structuur en geeft een lijst als return waarde met mensen van meest populair naar minst populair. (populair = gemeten door aantal vrienden dat je hebt). Als er personen zijn met dezelfde aantal vrienden dan moeten die alfabetisch worden gegeven.

def via_via_vrienden(sn, naam_persoon1, naam_persoon2):
deze functie bekijkt of persoon 1 op een of ander manier bevriend is met persoon 2 en geeft een boolean terug. Dit wil zeggen dat er een soort van "ketting" verbinding kan gemaakt worden tussen deze 2 mensen. Bijvoorbeeld A kent B, B kent C, C kent D, D kent E, dan moet via_via_vrienden(sn, A, E) True geven. (natuurlijk kan elk van deze ook meerdere vrienden hebben, er moet gewoon een duidelijke verbinding gemaakt worden.)