Examen Datastructuren en Algoritmen II

Naam : ....

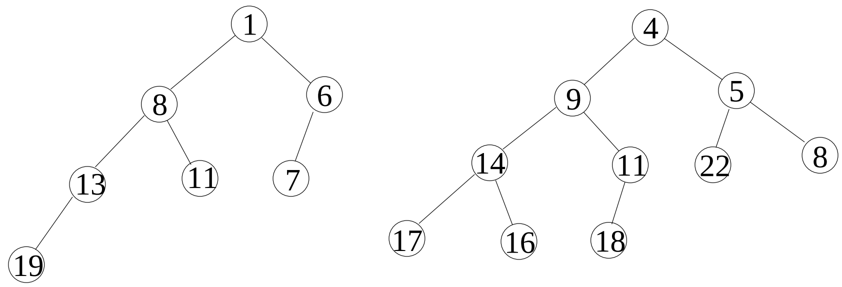
Kies 7 vragen uit deze 8. Er moeten maar 7 vragen beantwoord worden. In het geval dat er voor alle 8 vragen een antwoord wordt ingediend, telt vraag 1 niet mee.

1. **Skew heaps:**

   - Mogen sleutels meerdere keren in een skew heap voorkomen? Wat zou het probleem zijn als deze voorwaarde wordt veranderd?
   - Voeg de volgende twee reeksen van getallen aan twee initieel lege skew heaps toe: $10, 14, 18, 7, 5, 20$ en $3, 6, 9, 12, 15, 18, 21$. Merge deze twee bekomen heaps tenslotte.

2. **Leftist heaps:**

   Merge de volgende twee leftist heaps:



   Wat is het verschil tussen leftist heaps en skew heaps? Wat zijn de voordelen van leftist heaps in vergelijking met skew heaps en wat zijn de voordelen van skew heaps in vergelijking met leftist heaps?

3. **Dynamisch programmeren:**

   Een buis van lengte l moet in kleine stukken gesneden worden. Jammer genoeg zijn niet alle lengten nuttig, maar het is wel bekend welke lengten $l_1, \ldots, l_k$ nuttig zijn. De lengten $l_1, \ldots, l_k$ zijn veel kleiner dan l, maar ze zijn allemaal gehele getallen (bijvoorbeeld maten in *mm*). Natuurlijk eist ook elke snede een beetje ruimte. De vraag is of er een opdeling in stukken van nuttige lengte bestaat zodat de hele buis wordt gebruikt (stel dat de snede maat 1 heeft). Beschrijf een algoritme (bijvoorbeeld pseudocode) met dynamisch programmeren voor dit probleem en toon aan hoe het werkt voor $l = 25, l_1 = 2, l_2 = 3, l_3 = 6$.

4. **Geamortiseerde complexiteitsanalyse:**

   Hoeveel stappen zijn in het slechtste geval nodig om een element aan een binomiale prioriteitswachtlijn waarin al n sleutels zitten toe te voegen?

   Hoeveel stappen zijn er gemiddeld per toevoeging nodig om n elementen aan een initieel lege binomiale prioriteitswachtlijn toe te voegen?

   Toon aan dat jouw antwoorden juist zijn.

5. **Tonen dat iets arbitrair slecht presteert:**

   Noem een algoritme voor het TSP-probleem online, als het eerst een rondreis R_k door de steden $s_1, \ldots, s_k$ bepaalt en dan de stad s_{k+1} toevoegt doordat het een boog $\{v,w\}$ uit de rondreis R_k door de bogen $\{v,s_{k+1}\},\{s_{k+1},w\}$ vervangt. Toon aan dat er geen constante c bestaat, zodat voor alle gewogen complete grafen en inputreeksen $s_1,\ldots,s_n$ het algoritme gegarandeerd een rondreis vindt, die ten hoogste c keer zo lang is als de beste rondreis.

6. **Een probleem vertalen:**

   Gegeven een complete graaf met gewichten op de bogen. Stel dat wij een goed algoritme hebben om de kortste rondreis te bepalen, die elke top **tenminste** één keer bezoekt – noem dit probleem MTSP.

   Vertaal het gewone TSP-probleem waarbij elke top precies één keer moet bezocht worden naar het MTSP probleem. Schets dus een polynomiaal algoritme dat wanneer het toegepast wordt op een gewogen complete graaf G als resultaat een input voor MTSP oplevert. Deze input is dus een gewogen graaf G'. Het is hierbij vereist dat als wij de lengte l van een kortste MTSP-rondreis in G' kennen, wij ook de lengte van een kortste Hamiltoniaanse rondreis in G kennen (dat is: in constante tijd uit l kunnen berekenen). Toon aan, dat jouw algoritme aan de eisen voldoet.

7. **Spelstrategieën:**

   Nog een wisselgeldspel: Gegeven een bedrag b van eurocenten en een getal m. De regels zijn als volgt: In het begin liggen 0 cent op tafel. Speler X begint. Als een speler aan de beurt is en er ligt al een bedrag van $t < b$ cent op tafel, moet hij er precies één munt bijleggen zodanig dat nog steeds ten hoogste een bedrag van b op tafel ligt (1 euro telt natuurlijk als 100 cent, etc). **Maar:** de munten mag hij uit de portemonnee van zijn tegenstander nemen! Het spel is gedaan als er b cent op tafel liggen of als er m munten liggen. Als er op dit moment een bedrag van b cent ligt, is speler X gewonnen, anders speler Y. De winnaar mag alle munten houden en heeft dus zoveel gewonnen als hij zelf op tafel legde (omdat de munten van zijn tegenstander waren).

   Stel nu dat alleen maar munten van 1, 5 en 10 cent toegelaten zijn. Wat is de waarde van het spel met $b = 20$ en $m = 5$?

   Tip: eerst nadenken voordat je de boom tekent !

8. **Gerandomiseerde algoritmen:**

Gegeven een graaf met n toppen, waarin elke top graad tenminste $\log 4n$ heeft. Gezocht is een kleuring van de toppen met de kleuren rood een blauw zodat elke top tenminste één rode en tenminste één blauwe buur heeft.

Beschrijf een Las-Vegas algoritme dat na k pogingen (elke poging mag lineaire tijd vereisen) met kans tenminste $1-(1/2^k)$ zo'n kleuring heeft gevonden. Toon aan dat jouw algoritme deze eigenschap bezit.

Bestaat zo'n kleuring altijd als de graaf aan de voorwaarden voldoet? Leg uit waarom dit geldt of geef een tegenvoorbeeld.