

Examen Datastructuren en Algoritmen II

Naam :

1. **Verzamelingen:** (4 pt)

Stel dat je deelverzamelingen uit een universum van grootte n wil bepalen aan de hand van de geziene union-find algoritmen.

a.) Wat is de maximale diepte van een boom die je met $m < n$ union bewerkingen hebt opgebouwd waarbij je de wortel van de samengevoegde verzamelingen op een random manier kiest? Geef een voorbeeld voor dit slechtste geval.

b.) Wat is de maximale diepte van een boom die je met $m < n$ union by size bewerkingen hebt opgebouwd? Hier is geen voorbeeld voor het slechtste geval vereist maar wel een bewijs van de bovengrens.

c.) Wat is de maximale diepte van een boom die je met $m < n$ union bewerkingen hebt opgebouwd waarbij je de wortel van de samengevoegde verzamelingen op een random manier kiest maar path-compression toepast? Geef een voorbeeld voor dit slechtste geval.

d.) Doe de volgende bewerkingen op een universum dat de elementen $1, 2, \dots, 6$ bevat: union(1,2), union(3,4), union(4,5), union(1,3), union(1,6), find(2), find(6)

Gebruik 1 keer union by size en kies 1 keer de wortel van de samengevoegde verzamelingen toevallig maar pas path-compression toe.

2. **Rood-zwart bomen:** (2 pt)

a.) Geef de definitie van een rood-zwart boom.

b.) Geef een benedengrens en een bovengrens voor het aantal **rode** toppen op het pad van de wortel naar een NULL-pointer in een rood-zwart boom met n sleutels. Geef uitleg. Stellingen uit de les die gebruikt worden, moeten niet opnieuw bewezen worden.

c.) Voeg de sleutels 1, 10, 20, 30, 15, 12 in deze volgorde toe aan een rood-zwart boom. Toon voldoende tussenstappen die tonen wat er gebeurt.

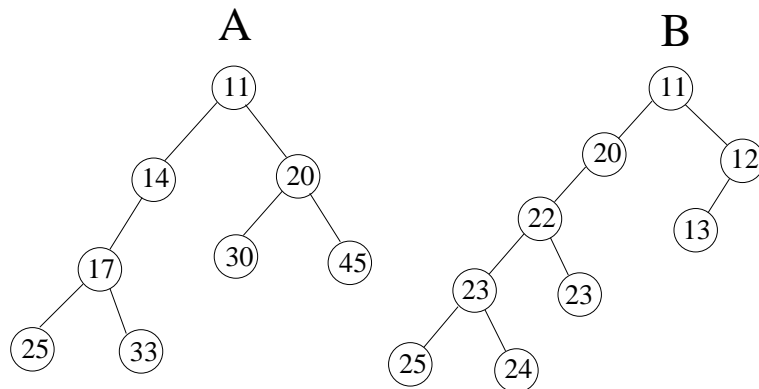
3. **Leftist heaps:** (2 pt)

a.) Wat is de maximale en wat de minimale lengte van een rechterpad in een leftist heap met n sleutels? (Geen bewijs vereist.)

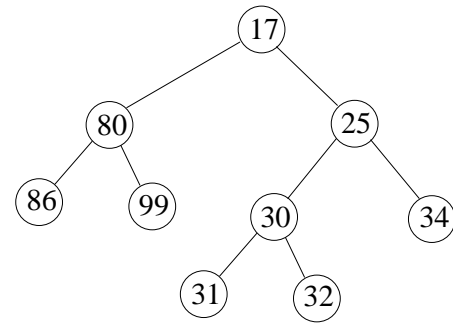
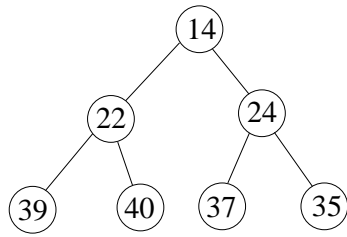
b.) Hoe duur is één toevoegbewerking op een leftist heap met n sleutels in het slechtste geval?

c.) Hoe duur is een **reeks** van n toevoegbewerkingen op een initieel lege leftist heap in het slechtste geval?

d.) Schrijf voor elk van de twee heaps of hij een leftist heap is en in het geval van niet waarom niet.



e.) Merge de twee volgende leftist heaps en toon voldoende tussenstappen om te laten zien wat er gebeurt.



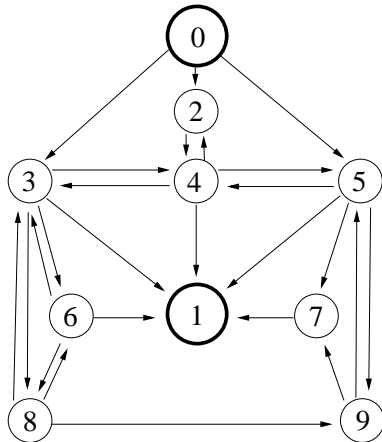
4. De waarde van een spel: (1 pt)

Een spel op een graaf: De spelers zijn X en O en ze kiezen afwisselend toppen. Elke top kan maar 1 keer gekozen worden. Speler X begint en **moet** top nummer 0 kiezen. Als net een top v werd gekozen dan moet de volgende top w een top zijn waarvoor een gerichte boog $v \rightarrow w$ bestaat (dus worden in principe gerichte paden gekozen). Het spel is gedaan als top 1 wordt gekozen (dan is de speler gewonnen die deze top heeft gekozen) of als geen top meer gekozen kan worden (dan is het gelijkspel en niemand wint iets).

In het geval dat een speler wint, is zijn winst de som van de van **hem** gekozen toppen. De andere speler moet hem deze winst betalen – dus is hij even veel verloren.

Als in de gegeven graaf de toppen 0, 3, 1 in deze volgorde worden gekozen, wint X dus $0 + 1 = 1$ cent en als 0, 3, 4, 2 in deze volgorde worden gekozen is de winst 0 voor beide spelers.

- Wat is de waarde van dit spel als je het op de volgende graaf speelt?
- Teken de spelboom en pas α - β -snoeien toe. Als je snoeit, geef uitleg waarom dat kan.
- Evalueer eerst de tak waar O voor top 2 kiest.



5. Gerandomiseerde algoritmen: (1 pt)

In AD III zullen jullie zien dat je niet alle bestanden kan comprimeren maar dat je inderdaad maar $1/255$ van alle mogelijke bestanden met een gegeven aantal bytes kan comprimeren. Voor deze oefening mag je daarvan uitgaan.

Als je het algoritme van gzip kent, kan je natuurlijk jouw kennis gebruiken om een bestand te vinden dat niet gecomprimeerd kan worden. Maar stel nu dat je een compressie-algoritme hebt waarvan je wel weet dat het heel snel is (b.v. in lineaire tijd een bestand kan comprimeren) maar niet wat het precies doet.

De taak is een efficiënt algoritme te beschrijven dat voor dit compressiealgoritme een niet comprimeerbaar bestand met n bytes kan vinden. Inderdaad lijkt hier een gerandomiseerd algoritme geschikt te zijn.

- Beschrijf een Las-Vegas algoritme voor dit probleem.
- Hoe groot is de kans dat dit algoritme meer dan 3 iteraties nodig heeft om het bestand te vinden?

6. Dynamisch programmeren: (2 pt)

Je wil precies n schroeven kopen, maar de schroeven kan je alleen maar in dozen met 1, 10, 25 of 100 schroeven kopen.

Geef een algoritme met dynamisch programmeren dat het minimum aantal dozen bepaalt dat je moet kopen om precies n schroeven te hebben.

7. **Een probleem vertalen:** (2 pt)

Vertaal het probleem met het minimale aantal dozen uit vraag 6 naar één van de volgende problemen uit de grafentheorie.

Kies één van de volgende problemen:

- TSP: wat is de kost van de goedkoopste rondreis in een gewogen complete graaf.
- Kortste gerichte paden: hoe lang is het kortste gerichte pad tussen twee gegeven toppen in een gerichte graaf.
- Goedkoopste paden: wat is de kost van het goedkoopste pad tussen twee gegeven toppen in een gewogen graaf.
- Taille: hoe lang is de kortste cykel in een graaf.
- Omvang van een gerichte graaf: hoe lang is de langste gerichte cykel in een graaf.
- Graad: wat is de graad van de top met de grootste graad in een graaf.

Vertalen betekent hier natuurlijk vooral de graaf te beschrijven waarvoor je het nieuwe probleem wil oplossen. De vertaling moet in tijd $O(p(n))$ mogelijk zijn, waarbij $p()$ een polynoom is. Geef ook uitleg waarom de vertaling juist is.

8. Geamortiseerde complexiteitsanalyse: (2 pt)

Stel dat voor binomiale wachtlijnen de kost van een toevoegbewerking gegeven is door het aantal bomen dat verenigd moet worden plus 1 voor het aanmaken van een boom met 1 element die het nieuwe element bevat.

a.) Hoe duur is een reeks van n toevoegbewerkingen aan een binomiale wachtlijn die in het begin leeg is. De $O()$ notatie is voldoende.

b.) Bewijs dat jouw antwoord op a.) juist is. Gebruik ofwel de accounting methode ofwel de potentiaal methode.

NOG NIET OMDRAAIEN !