Examen Datastructuren en Algoritmen II

Naam : ....

Lees de hele oefening zorgvuldig voordat je begint ze op te lossen!
Als je niet goed verstaat wat de vraag of taak is, vraag het aan de lesgever!
Voor oefeningen die fout verstaan zijn, kunnen geen punten gegeven worden!

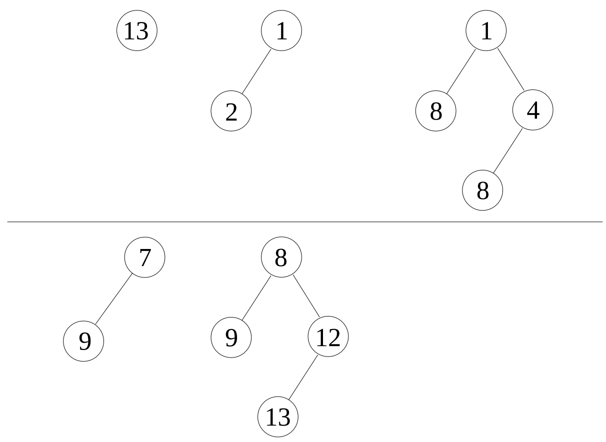
Schrijf leesbaar. Oplossingen die niet leesbaar zijn, kunnen ook niet beoordeeld worden.

Als technieken toegepast moeten worden, toon altijd voldoende tussenstappen om te kunnen zien wat er gebeurt en dat de technieken goed verstaan zijn.

Stellingen uit de les mogen natuurlijk altijd gebruikt worden zonder dat het bewijs opnieuw gegeven moet worden (behalve in gevallen waar het expliciet anders staat)!

1. **Wachtlijnen 4 pt**

   - Merge de volgende twee binomiale prioriteitswachtlijnen. Verwijder achteraf het kleinste element. Als er in twee binomiale bomen hetzelfde kleinste element in de wortel zit, verwijder dan het element uit de grotere boom (in de praktijk zou dat misschien niet de beste oplossing zijn).



- Voeg de sleutels 11, 19, 15, 30, 17, 5 en 13 in deze volgorde toe aan een initieel lege leftist heap. Verwijder achteraf het kleinste element.

- Wat is de maximale en minimale **diepte** van een leftist heap? Geef een redenering die aantoont dat jouw grenzen juist zijn en geef reeksen $s_1, \ldots, s_n$ zodat als de sleutels in deze volgorde aan een leftist heap toegevoegd worden de heap deze maximale resp. minimale diepte heeft.

- Wat is de maximale en minimale lengte van een rechterpad in een leftist heap? Geef een redenering die aantoont dat jouw grenzen juist zijn. Daarbij mogen stellingen uit de les zonder bewijs geciteerd worden. Geef reeksen $s_1, \ldots, s_n$ zodat als de sleutels in deze volgorde aan een leftist heap toegevoegd worden de heap deze maximale resp. minimale lengte van het rechterpad heeft.

- De lengte van het rechterpad in een skew heap met n elementen kan $\Omega(n)$ zijn. Maar hoe lang kan het precies zijn? Geef en bewijs de beste grens die je kan bewijzen. De volle punten worden alleen gegeven voor het bewijs dat de lengte (het aantal bogen op het rechterpad) **ten hoogste** $\frac{n-1}{2}$ is.

2. **Zoekbomen 3 pt**

   - Voeg de sleutels 28, 13, 5, 30, 1, 21, 27, 33 in deze volgorde toe aan een rood-zwart boom. Gebruik – waar mogelijk – de geoptimaliseerde bewerkingen die ervoor zorgen dat geen nieuw conflict kan ontstaan.

- Gegeven een semi-splay boom T en een top v in T op afstand d van de wortel. Wat is de grootst mogelijke afstand die v na een volgende opzoekbewerking kan hebben (als functie van d). De opzoekbewerking kan – maar hoeft niet – van v te zijn. Bewijs dat jouw antwoord juist is.

- Geef grenzen voor de maximale en minimale diepte van een 2-3 boom met n toppen. Bewijs dat jouw grenzen juist zijn.

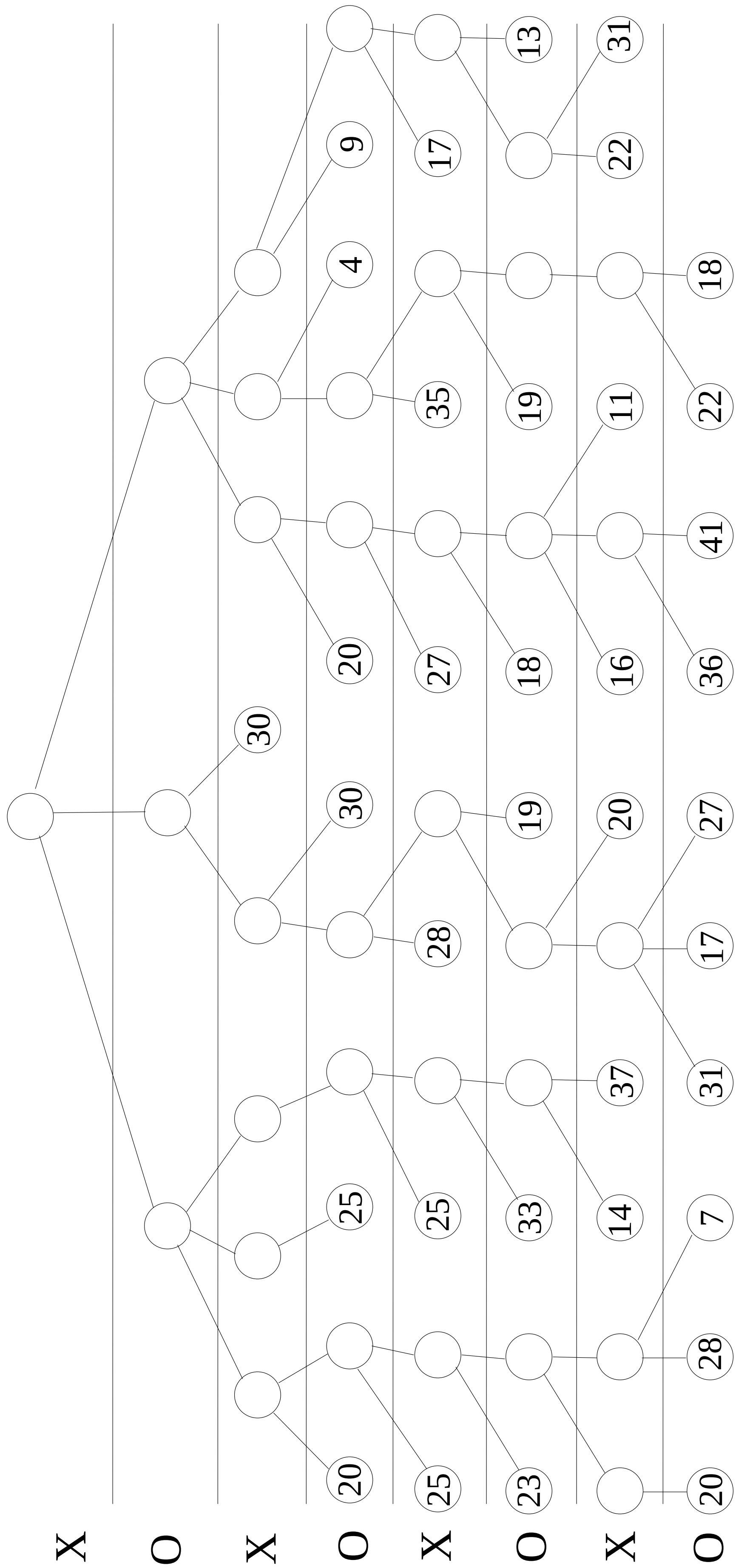
3. **Online algoritmen 1 pt**

   Bewijs de volgende stelling:

   Voor elk online inpakalgoritme en elke $k \in \mathbb{N}$ bestaat er een inputreeks $g_1, \ldots, g_n$ met $n \geq k$ zodat het algoritme ten minste $\frac{3}{2}m$ vrachtwagens gebruikt waarbij m het aantal vrachtwagens in een optimale oplossing is.

4. **1.5 pt**

Pas α–β–snoeien op de volgende spelboom toe om de waarde van het spel te berekenen. Evalueer altijd eerst de linkertakken. Schrijf gebruikte grenzen **aan** de toppen en de teruggegeven waarden **in** de toppen. Plaats streepjes door de bogen die naar deelgrafen leiden die je niet evalueert omdat je snoeit.



5. **Geamortiseerde complexiteitsanalyse 3 pt**

- In de les hebben wij gezien dat een reeks van n bewerkingen op een initieel lege semi-splay boom een kost van $O(n\log n)$ heeft (dus geamortiseerd $O(\log n)$ per bewerking). Stel nu dat de boom in het begin niet leeg is, maar dat er al n^2 bewerkingen gebeurd zijn. Voor deze n^2 bewerkingen was de boom wel leeg. Bewijs dat dan de kost van de reeks bestaande uit de laatste n bewerkingen niet meer $O(n\log n)$ is.

- Je werkt met een verzameling van twee binaire hopen, die ook leeg kunnen zijn. De mogelijke bewerkingen zijn als volgt:

Toevoegen: Als één van de twee hopen kleiner is dan de andere wordt het nieuwe element aan de kleinere hoop toegevoegd. De kost is dan $\log k$ met k het aantal elementen in de kleinere hoop.
Als beide hopen dezelfde grootte g hebben dan worden zij en het nieuwe element samengevoegd tot één grote hoop met $2g+1$ elementen en de tweede hoop in de verzameling is leeg. De kost van deze bewerking is $2g+1$.

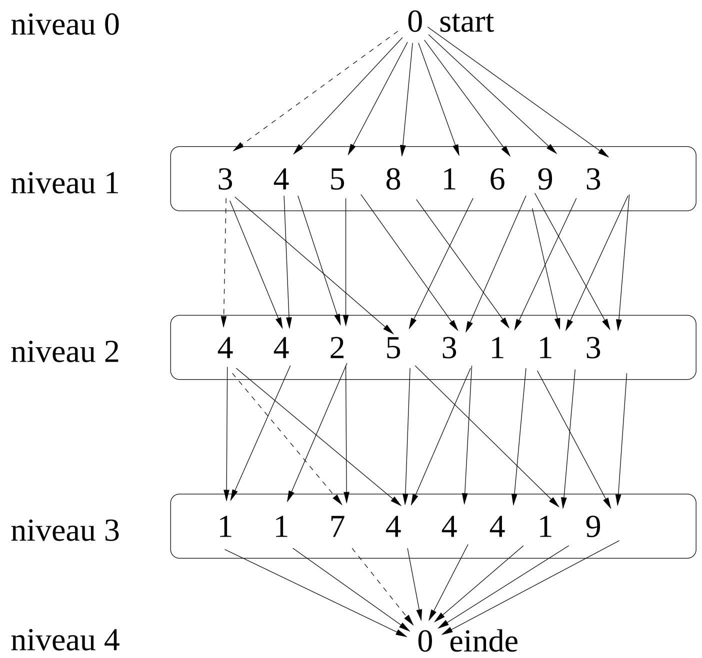
Uitvoeren: Alle op dit moment aanwezige elementen worden in volgorde uitgevoerd. Achteraf bestaat de verzameling uit twee lege hopen. De kost van deze bewerking is $k * \log k$, met k het aantal elementen in de twee hopen samen als de hopen niet leeg zijn en anders 1.

Bepaal de maximale en de geamortiseerde kost van een bewerking in een reeks van n bewerkingen op een initieel lege verzameling. De $O()$-notatie is voldoende. Gebruik de methode die je het meest geschikt lijkt. Probeer het best eerst of minder moeilijke methoden al de gewenste resultaten opleveren.

6. **Een algoritme kiezen 2.5 pt**

Gegeven een gewogen gerichte graaf $D = (V, E)$. D bevat een startpunt met waarde 0 (op niveau 0) en een eindpunt met waarde 0 op niveau $m+1$. Alle andere toppen zitten op niveaus $1, \ldots, m$ en hebben gehele waarden $w() \geq 0$. Op elk niveau zitten k toppen. De bogen verlopen altijd van een top op niveau l, $0 \leq l \leq m$, naar niveau $l+1$ maar alle dergelijke bogen zijn niet noodzakelijk in E aanwezig.

De spelers X en O beginnen nu vanaf het startpunt een pad op te bouwen. Elke speler kiest een boog die in het eindpunt van het al opgebouwde pad vertrekt (in het begin het startpunt) en één niveau dieper gaat totdat het eindpunt bereikt is. De waarde van het pad is de som van de waarden van de toppen op het pad. Als de som even is moet O dit bedrag (in Eurocent) aan X betalen en anders betaalt X het bedrag aan O. Als het met streepjes gemarkeerde pad in het voorbeeld gekozen wordt, wint X dus 14 Eurocent.



Beschrijf een algoritme om de waarde van het spel te berekenen dat polynomiaal is in de grootte van de gegeven graaf. Geef het algoritme en leg uit waarom het polynomiaal begrensd is.

7. **Gerandomiseerde algoritmen 1 pt**

   Je werkt in de tweede selectie van de kwaliteitscontrole van een bedrijf dat televisietoestellen maakt. De televisietoestellen worden in dozen van 75 naar de groothandelaars gestuurd en van de toestellen die de tweede selectie bereiken, is gemiddeld 1 op 1000 nog defect. Het doel is nu zo weinig mogelijk toestellen te controleren maar toch te garanderen dat een doos die een fout toestel bevat een winkel bereikt ten hoogste 5% is.

   Geef een algoritme dat aan de eisen voldoet en zo weinig mogelijk toestellen test. Beschrijf het algoritme en toon aan dat het aan de eisen voldoet. Als je een formule voor een getal a hebt van de vorm bv. $f(a) \leq c$ moet je a niet echt bepalen – het is voldoende te zeggen "kies a minimaal/maximaal zo dat het aan deze formule voldoet".

NOG NIET OMDRAAIEN !