

Examen Datastructuren en Algoritmen II

Naam :

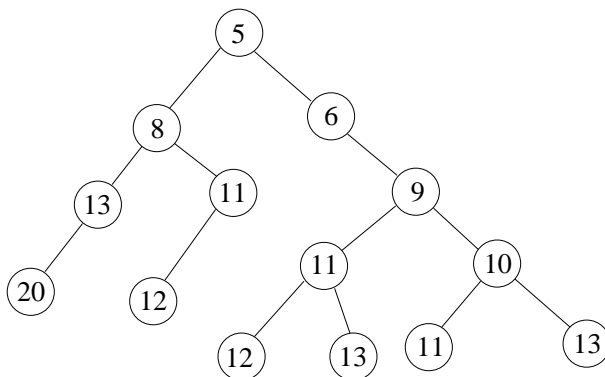
1. Skew-heaps (2 pt)

In deel a.) en b.) is de $O()$ notatie voldoende.

a.) Hoe duur is één toevoegbewerking op een skew-heap met n toppen in het slechtste geval ? (Geen bewijs vereist.)

b.) Hoe duur is een reeks van n toevoegbewerkingen op een initieel lege skew-heap in het slechtste geval ? (Geen bewijs vereist.)

c.) Verwijder het kleinste element uit de volgende skew-heap. Toon voldoende tussenschappen om te laten zien wat er gebeurt.

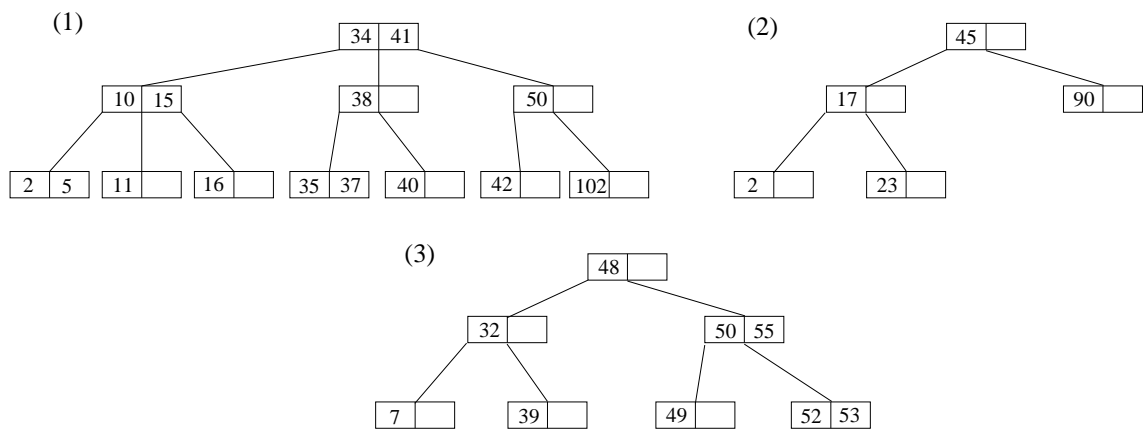


2. 2-3-bomen: (3 pt)

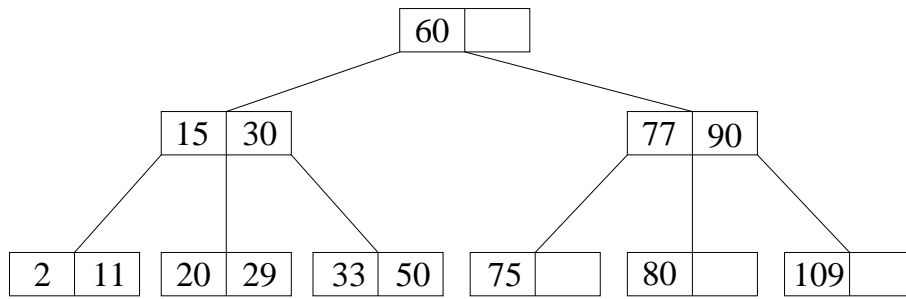
a.) Wat is in het slechtste geval de diepte van een 2-3-boom met n sleutels en wat is de diepte in het beste geval? Wat zijn de beste en slechtste gevallen? (Geen bewijs vereist.)

b.) Hoeveel vergelijkingen van sleutels zijn er in het beste geval nodig om een nieuwe sleutel aan een 2-3-boom met n sleutels toe te voegen? Wat is het beste geval? (Geen bewijs vereist.)

c.) Schrijf voor elk van de drie bomen of hij een 2-3-boom is en in het geval van niet waarom niet.



d.) Voeg de sleutel 24 toe aan de volgende 2-3-boom. Toon voldoende tussenstappen om te zien wat er gebeurt, maar voor de tussenstappen moet je niet elke keer de **hele** boom tekenen.



3. **Het handelsreizigersprobleem:** (2 pt)

a.) Wat is een beslissingsprobleem?

b.) In de les hebben wij gezien dat er meerdere formuleringen van het handelsreizigersprobleem zijn. Geef één formulering als een beslissingsprobleem en één formulering die geen beslissingsprobleem is.

c.) Wij hadden gezegd dat alle formuleringen van het handelsreizigersprobleem *equivalent* zijn. Wat betekent *equivalent* in deze samenhang?

d.) Geef een branch-and-bound algoritme (b.v. in pseudocode) voor één van jouw formuleringen van het handelsreizigersprobleem. Er moet wel één bounding criterium inzitten maar het mag wel een gemakkelijk criterium zijn.

4. Gerandomiseerde algoritmen: (2 pt)

Een computerprogramma voor een beslissingsprobleem dat bepaalt of een geheel getal een zekere eigenschap heeft (bijvoorbeeld priem is – maar het kan ook een andere eigenschap zijn) wordt door een foute compiler gecompileerd en de uitvoerbare bestanden met deze foute programma's worden ook via het internet verdeeld. Maar er zijn ook kopieën van dit programma die juist zijn. Je zit nu aan een computer en moet beslissen of het programma betrouwbaar is. Je weet dat de foute versies van het programma voor een kwart van de getallen **soms** een fout antwoord geven. Ook voor de getallen waar iets misloopt is de kans een juist antwoord te krijgen nog altijd $2/3$. Beschrijf een algoritme om met kans ten minste 99% te weten of het programma waarmee je moet werken juist is.

Let op: Je hebt geen lijst van juiste antwoorden ter beschikking!

5. **Tonen dat iets slecht presteert: Het online inpak probleem** (2 pt)

- a.) Geef een voorbeeld waar first-fit beter presteert dan best-fit en een voorbeeld waar best-fit beter presteert dan first-fit.
- b.) In de les hebben wij gezien op welke manier je voor een gegeven online algoritme A voor elk getal m een reeks kan vinden die ten minste m vrachtwagens vraagt en het algoritme A ten minste $4/3$ keer het optimum aantal vrachtwagens gebruikt. Als het toegelaten is dat de reeksen altijd maar één of twee vrachtwagens eisen kan dat nog beter:

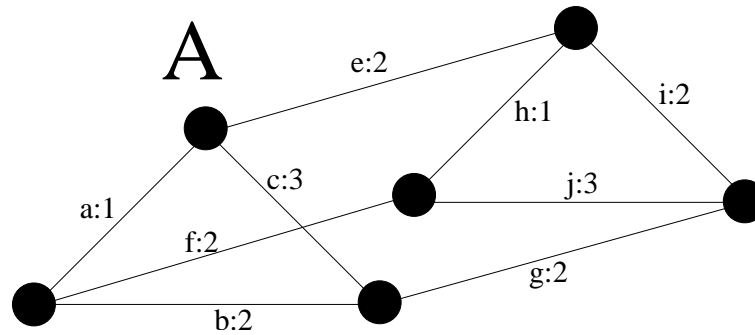
Toon aan dat voor elk online algoritme en elk getal g een reeks met ten minste g gewichten bestaat waarvoor het algoritme ten minste $3/2$ keer het optimale aantal vrachtwagens gebruikt.

Tip: Geef een reeks die op twee vrachtwagens kan geplaatst worden en die een deelreeks bevat die op één vrachtwagen kan geplaatst worden. De structuur moet zo zijn dat als de deelreeks op één vrachtwagen wordt geplaatst de hele reeks 3 vrachtwagens nodig heeft.

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

Een spelletje op een graaf waar de gewichten op de bogen centen zijn: Je begint met een zekere top A en speler X kiest eerst. Je kiest altijd toppen die buren van de laatst gekozen top zijn en gaat langs de boog naar de nieuwe top. Je mag alleen maar toppen kiezen die nog niet gekozen zijn behalve in één geval: Als alle buren van de laatst gekozen top al gekozen zijn en de starttop is een buur, dan mag je ook de starttop kiezen. Het spel is gedaan als geen buur meer kan gekozen worden en de winst is de som van de gewichten van de bogen waar de spelers langs zijn gegaan. Als het spel stopt en de gekozen bogen vormen een hamiltoniaanse cykel is de winst voor X (en Y moet het betalen) – anders is de winst voor Y (en X moet het betalen).

Wat is de waarde van het spel voor de volgende graaf? Teken de spelboom. Als je snoeit geef de reden. De letters aan de bogen zijn er gewoon om de spelboom beter te kunnen tekenen. Bereken eerst de waarde in de deelboom waar X boog e kiest.



Dit is de vervangoefening voor de huiswerk oefeningen. Ze staat op 4 punten. Voor het examen wordt het maximum van de in deze oefening verworven punten en de punten uit de huiswerk oefening geteld. Wie in de huiswerk oefeningen al goed presteerd heeft zou deze oefening dus het best ofwel niet bekijken ofwel alleen maar als hij/zij met alle andere oefeningen klaar is.

7. Geamortiseerde complexiteitsanalyse: Een array groter maken: (4 pt)

Een bekende manier om met een array te werken is dat je elke keer dat je een element wil toevoegen dat niet meer in de array past dubbel zo veel ruimte voor de array allocceert en de oude array kopieert.

Maar je moet wel veilig stellen dat de nieuwe array goed geïnitieerd is. Wij stellen dat dat één stap vraagt per plaats die geïnitieerd moet worden.

- a.) Hoeveel stappen vraagt een toevoeg- of verwijder-bewerking in het slechtste geval?
Let op: een toevoeg-bewerking kan een uitbreiden van de array als gevolg hebben – die kosten moet je natuurlijk meetellen!
- b.) Toon aan dat een reeks van n toevoeg- en verwijder-bewerkingen op een initieel lege array $O(n)$ stappen vraagt (gebruik ofwel de potentiaal- ofwel de accounting-methode).

Stel nu dat je om ruimte te besparen bovendien elke keer dat de array niet helemaal leeg is maar toch minder dan $1/4$ gevuld een nieuw array met maar de halve grootte allocceert en gebruikt.

- c.) Toon aan dat een reeks van n toevoeg- en verwijder-bewerkingen op een initieel lege array nog altijd $O(n)$ stappen vraagt. Als je voor b.) de potentiaal-methode hebt gebruikt, gebruik nu de accounting-methode – anders gebruik de potentiaal-methode.

Let op: Er is een belangrijk verschil met de huiswerk oefening die op deze oefening lijkt. Maar dit verschil maakt de oefening gemakkelijker. . .

NOG NIET OMDRAAIEN !