

Examen Datastructuren en Algoritmen II

Naam :

Lees elke oefening zorgvuldig en helemaal voordat je met de oefening begint!

1. (3 pt)

- Het universum is $\{0, 1, \dots, 7\}$ en de gegeven relaties zijn $0 \equiv 1$, $2 \equiv 3$, $4 \equiv 5$, $6 \equiv 7$, $1 \equiv 2$, $5 \equiv 2$, $0 \equiv 6$, $6 \equiv 2$.

Bereken de equivalentieklassen van de gegenereerde equivalentierelatie door union-by-size en path compression toe te passen. In gevallen waar door deze regels niet vastgelegd wordt welke van de twee toppen de wortel wordt, neem die met de grotere sleutel.

- Stel dat je een universum met n elementen hebt en je doet $m < n$ union- of find-bewerkingen door union-by-size en path compression toe te passen. Wat is de maximale **diepte** van één van de bomen die een verzameling voorstellen? De $O()$ -notatie is niet voldoende. Geef uitleg.

- Stel dat jouw universum $\{0, 1, \dots, n\}$ is en voor $0 \leq i < n$ pas je de relaties $i \equiv i + 1$ toe (beginnend met $i = 0$). Je past het union-algoritme zonder union by size of path compression toe en kiest altijd de wortel met de grotere sleutel als nieuwe wortel. Wat is de geamortiseerde kost van deze reeks van bewerkingen. Geef uitleg.

- Stel dat jouw universum $\{0, 1, \dots, n\}$ is en voor $1 \leq i \leq n$ pas je de relaties $0 \equiv i$ toe (beginnend met $i = 1$). Je past het union-algoritme zonder union by size of path compression toe en kiest altijd de wortel met de grotere sleutel als nieuwe wortel. Wat is de geamortiseerde kost van deze reeks van bewerkingen. Geef uitleg.

2. (3 pt)

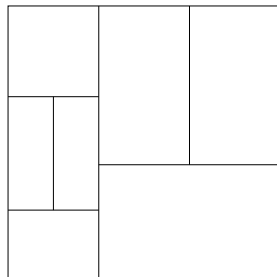
- Voeg de sleutels 20,10,7,22,1,12,19,30 in deze volgorde toe aan een rood-zwart boom. Gebruik de in de les gezien bewerkingen om de boom te herbalanceren.
Toon voldoende tussenstappen om te zien wat er gebeurt.

- Bewijs gedetailleerd dat voor elke datastructuur die een binaire zoekboom implementeert een reeks van n toevoegbewerkingen op een initieel lege boom bestaat zodat de kost (het aantal bezochte toppen) van de hele reeks van toevoegbewerkingen ten minste $\Omega(n \log n)$ is.

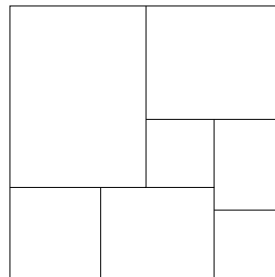
3. Hier kan je kiezen. Kies **precies één** van de volgende twee oefeningen. Oefening (b) is gemakkelijker maar je kan daarmee ook minder punten halen – je geeft dan 1 punt op. Kies dus alleen voor (b) als je (a) te moeilijk vindt!
-

((a) 3 pt)

In een bedrijf moet een kwadratisch stuk koper met grootte $l \times l$ opgedeeld worden in verschillende rechthoekige stukken van lengte $d \times d'$ met gehele lengten d en d' . Voor elk paar d, d' is de prijs $p(d, d')$ die voor het rechthoek met grootte $d \times d'$ gekregen kan worden gekend. Het doel is nu het stuk koper zo in stukken te zagen dat de totale prijs maximaal is. **Maar:** Als je zaagt moet je altijd door het hele stuk dat je zaagt zagen – dat geldt voor het hele kwadraat en voor de rechthoeken die je na een snede hebt! De breedte van de snede is 1. Beschrijf een dynamisch programmeren algoritme in pseudocode en geef uitleg over jouw code indien nodig.



mogelijk



niet mogelijk

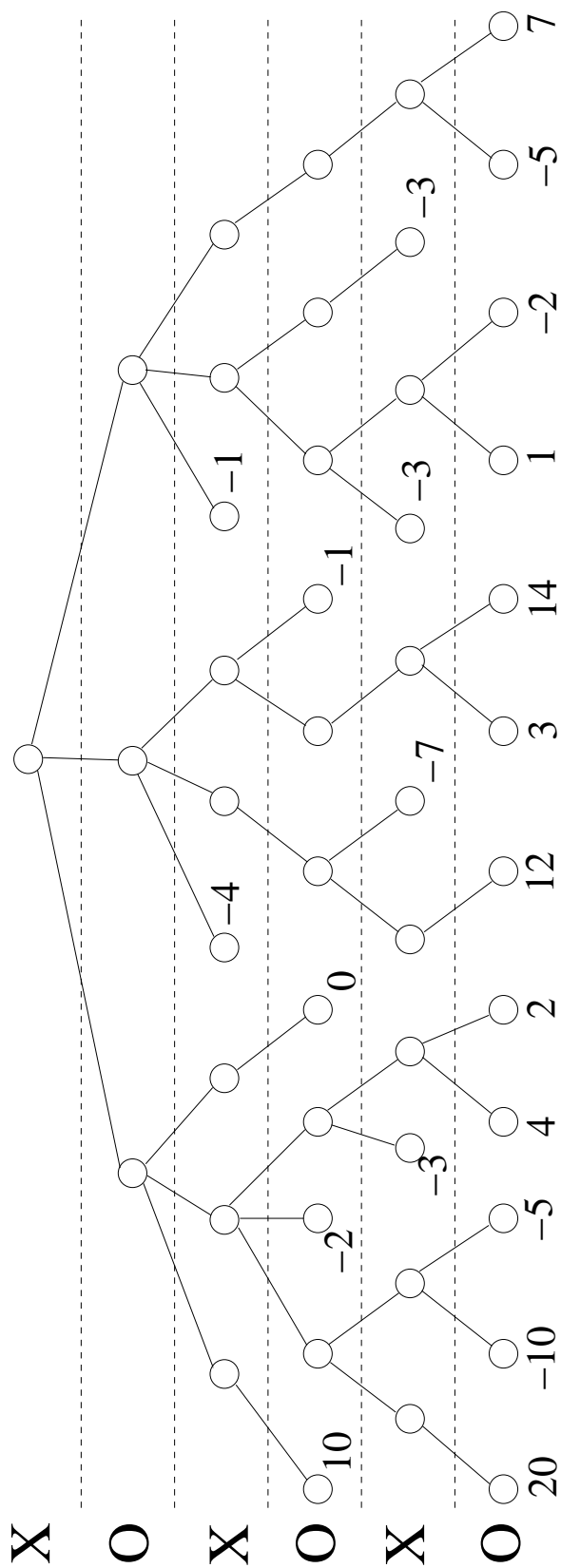
((b) 2 pt)

In een bedrijf moet een kwadratisch stuk koper met grootte $l \times l$ opgedeeld worden in verschillende rechthoekige stukken van lengte $d \times l$ met gehele lengten d . Voor elke lengte $d \leq l$ is de prijs $p(d)$ die voor het rechthoek met grootte $d \times l$ gekregen kan worden gekend. Het doel is nu het stuk koper zo in stukken te zagen dat de totale prijs maximaal is. De breedte van de snede is 1. Beschrijf een dynamisch programmeren algoritme in pseudocode en geef uitleg over jouw code indien nodig.

4. (2 pt)

- Geef de definitie van de waarde van een spel. Zeg daarbij ook expliciet wat er gegeven moet zijn.

- Bereken de waarde van deze spelboom. Pas α - β -snoeien toe waar mogelijk en schrijf de waarden en de grenzen van de waarden aan de toppen waar je ze berekend hebt. Bij het doorlopen van de boom neem altijd eerst de linkertak.



5. (1.5 pt)

Gegeven een graaf $G = (V, E)$ en $V' \subseteq V$. Als $E' \subseteq E$ de verzameling van alle bogen is waarvan beide toppen in V' liggen dan heet $|E'|/|V'|$ de dichtheid van de door V' geïnduceerde deelgraaf.

Geef een branch and bound algoritme voor het probleem de dichtste deelgraaf van een graaf te vinden. Geef daarbij ten minste één niet triviaal bounding criterium. Beschrijf het algoritme in pseudocode en geef uitleg indien nodig.

6. (1.5 pt)

In een bedrijf dat rondreizen die zekere eigenschappen hebben (en door n gegeven steden gaan) voor een klant moet berekenen wordt een algoritme toegepast dat recursief alle $n!$ mogelijke volgorden test. Wat de eigenschappen zijn is niet zo belangrijk en die wisselen ook vaak.

Een branch and bound algoritme is niet mogelijk omdat de eisen waaraan de rondreizen voldoen altijd veranderen, maar volgens de ontwikkelaars ook niet nodig omdat ze berekend hebben dat als je naar de call-tree van de recursie kijkt voor alle mogelijke eisen die gebruikt worden ten minste 10% van de bladeren rondreizen met de gewenste eigenschappen zijn – het is dus niet echt een ingewikkeld probleem. Maar toch is het zo dat de computers soms moeilijkheden hebben een rondreis te vinden en na uren nog altijd doorgaan...

Geef advies aan de ontwikkelaars van dit bedrijf hoe ze hun probleem kunnen oplossen. Geef een goede argumentatie waarom jouw oplossing goed is.

7. (2 pt)

- Bewijs de volgende uitspraak of geef een tegenvoorbeeld:

Er bestaat een constante C zodat het Best-Fit online algoritme ten hoogste C vrachtwagens meer gebruikt dan het optimum.

- Toon aan:

Er bestaat een polynomiaal algoritme dat voor een gegeven reeks g_1, \dots, g_m van gewichten een oplossing kan vinden om die op n vrachtwagens met capaciteit 1 te plaatsen (als een dergelijke oplossing bestaat) als en slechts als er ook een polynomiaal algoritme bestaat dat voor een gegeven reeks c_1, \dots, c_n van capaciteiten en een gegeven reeks g_1, \dots, g_m van gewichten een manier kan vinden om de gewichten op de vrachtwagens te plaatsen als zo'n manier bestaat.

Tip: Eén richting is triviaal. Voor de andere herschaal de gewichten en werk met n dummygewichten. Je moet de problemen dus vertalen en deze vertaling moet in polynomiale tijd berekenbaar zijn.

NOG NIET OMDRAAIEN !