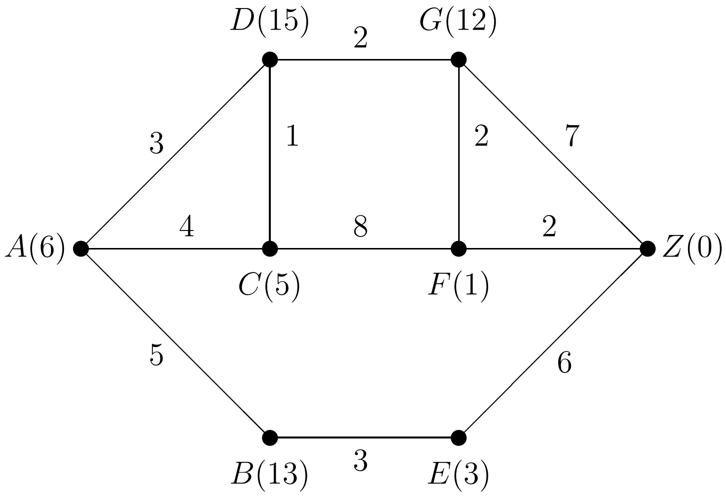
Tentamen Kunstmatige intelligentie
Universiteit Leiden — Informatica
maandag 12 augustus 2002, 13.00–16.00 uur

Geef bij alle antwoorden een korte en duidelijke toelichting. Veel succes!

Opgave 1. A*/IDA* (20 punten)

a. (6 punten) Leg het A*-algoritme en het IDA*-algoritme uit.
b. (2 punten) Noem een situatie waarin IDA* bijzonder slecht presteert.
c. (2 punten) Wanneer is een heuristische functie *admissibel* (= *toelaatbaar*)?
d. (2 punten) In onderstaande graaf is de beginknoop A, en de doelknoop Z. De kostenfunctie staat naast de takken. Bij de knopen staat tussen haakjes de heuristische functie. Vervang op de plaatsen waar deze functie niet admissibel is de waarde door 0.
e. (8 punten) Voer zowel het A*-algoritme als het IDA*-algoritme uit voor deze graaf. Gebruik zonodig de *pathmax equation*. Geef duidelijk aan hoe de algoritmes verlopen.



Opgave 2. α-β-algoritme (20 punten)

We bekijken het volgende spel voor twee personen. Het speelt zich af op een in eerste instantie leeg drie bij drie bord. Om en om, te beginnen met `W`, zetten de spelers `W` en `Z` een eigen schijf op een tot dan toe leeg vakje. Na twee zetten kan het er zo uit zien:

```
W . .
. Z .
. . .
```

De speler die de meeste vakjes kan "zien" (als een dame in het schaakspel: horizontaal, verticaal en diagonaal) wint, en hoe meer verschil des te beter. Zijn het er evenveel, dan is het spel remise. In het voorbeeld ziet `W` er 4 en ziet `Z` er 7, en dus wint `Z`. Je kunt niet door andere schijven heen kijken; vakjes waar een schijf staat tellen niet mee.
a. (5 punten) Geef de *spelboom* (= *game tree*) die hierbij hoort. Neem aan dat beide spelers één maal aan de beurt komen. Gebruik zoveel mogelijk symmetrie!
b. (5 punten) Beschrijf in woorden het *minimax-algoritme*.
c. (5 punten) Voer dit uit voor de boom van **a**. Maak zelf een geschikte evaluatiefunctie.

d. (5 punten) Voer het *α-β-algoritme* uit voor de spelboom, en geef duidelijk aan waar gesnoeid (= "gepruned") wordt. Geef ook een korte rechtvaardiging voor dit snoeien. Zorg ervoor dat de ordening van de knopen zo is dat er *zoveel mogelijk* gesnoeid kan worden.

Opgave 3. ID3 (20 punten)

We hebben een database waarin een aantal voorbeelden aan de hand van een stel attributen beschreven worden; verder hebben we een Ja/Nee-classificatie voor deze voorbeelden.
a. (10 punten) Leg uit hoe het *ID3-algoritme* werkt. Geef hierbij de definitie van de entropie $I(p/(n+p), n/(n+p))$ (p is het aantal positieve voorbeelden, n het aantal negatieve). Geef met name aan wat er moet (kan) gebeuren in de *vier* verschillende "standaard"-gevallen, bijvoorbeeld als er bij het splitsen geen voorbeelden meer zijn.
b. (5 punten) De volgende database geeft situaties uit het verleden aan. We willen een beslissingsboom maken om te kijken of een film goed of slecht is, op grond van de acteurs en de lengte. Maak de twee mogelijke beslissingsbomen met hoogte 2.

dag	acteurs	lengte	film goed?
1	slecht	kort	ja
2	goed	lang	ja
3	gaat	lang	ja
4	slecht	kort	nee
5	gaat	kort	nee
6	goed	kort	ja

c. (5 punten) Reken voor de boom met als wortelvraag de vraag naar de acteurs, de entropie-winst ("gain") uit voor deze eerste vraag.

Opgave 4. Genetische algoritmen (20 punten)

Stel dat we een handelsreizigersprobleem op willen lossen, en dat we dat met een *genetisch algoritme* willen doen. Gegeven zijn steden en al hun onderlinge afstanden, gezocht een kortste route die alle steden bezoekt, en eindigt waar de route begonnen is.
a. (5 punten) Geef (in woorden) aan hoe een genetisch algoritme er in het algemeen uit ziet.
b. (5 punten) Geef (in woorden) aan hoe een *evaluatiefunctie* (= *fitnessfunctie*) er in dit speciale geval uit zou kunnen zien.
c. (5 punten) Geef aan hoe *crossover* en *mutatie* in dit speciale geval zouden kunnen werken.
d. (5 punten) Wanneer heet een genetisch algoritme *elitair*?

Opgave 5. Data mining (20 punten)

We bekijken een transactie-database met verkoopgegevens over de producten (= items) 1,2,3,4,5,6,7.
a. (5 punten) Geef de definitie van *support* en *frequente itemset*.
b. (5 punten) Op welke belangrijke eigenschap van itemsets is het *Apriori-algoritme* voor het vinden van de frequente itemsets gebaseerd? Leg deze eigenschap kort uit.
c. (5 punten) Leg uit hoe het Apriori-algoritme werkt.
d. (5 punten) Stel dat het Apriori-algoritme de volgende frequente itemsets met 2 items gevonden heeft: $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, $\{2,4\}$, $\{2,7\}$, $\{3,4\}$, $\{3,5\}$, $\{3,7\}$ en $\{4,5\}$. Welke itemsets met 3 items worden allereerst gegenereerd en welke daarvan worden toch verworpen? En zijn de resterende sets automatisch frequent? Leg uit.