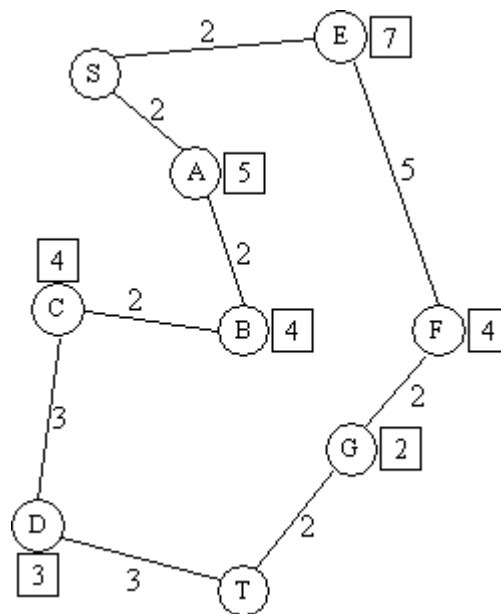


# AI Kaleidoscoop, Deeltentamen 1, 25 Oktober 2000

**Puntenverdeling:** (Je eindcijfer is je puntenscore gedeeld door 10)

|    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|
| 1a | 1b | 1c | 2a | 2b | 3a | 3b | 3c | 4a | 4b | 4c |
| 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 5  | 5  |

## Vraag 1: Heuristische zoekmethoden



Gegeven is de bovenstaande graaf. Het doel is om het kortste pad te vinden vanaf de startlocatie  $S$  naar de eindlocatie  $T$ . Bij elke locatie staat steeds in een vierkantje de geschatte afstand vanaf die locatie tot aan de eindlocatie. Langs elke kant staat steeds de daadwerkelijke lengte van die verbinding (die we natuurlijk pas weten als we die afstand ook echt hebben afgelegd).

**Vraag a:** Teken de zoekboom die ontstaat als we bovenstaande gegevens gebruiken om met behulp van het A\* algorithm voorwaarts een pad te zoeken van  $S$  naar  $T$ .

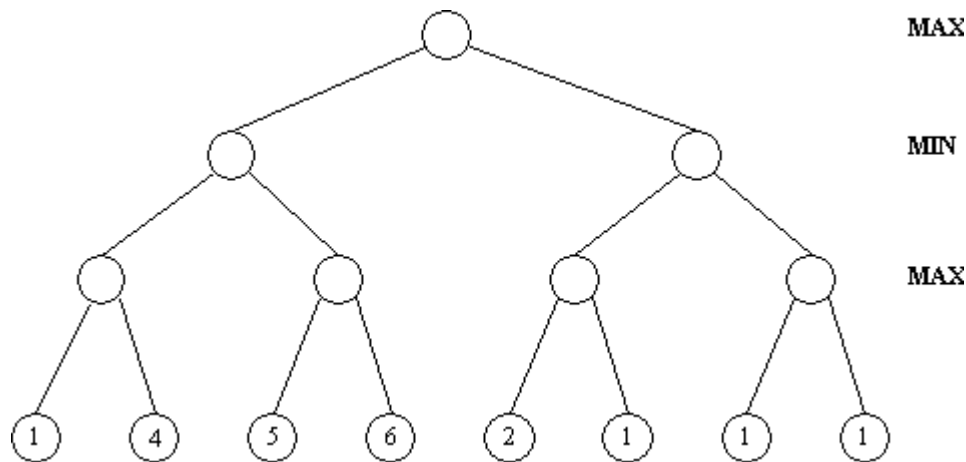
- Nummer de knopen in de zoekboom om aan te geven in welke volgorde ze door A\* bezocht worden.
- Geef ook bij elke knoop in de boom aan wat de heuristische waarde van die knoop is
- Schrijf op wat het uiteindelijke pad is dat door het A\* algorithm wordt gevonden om van  $S$  naar  $T$  te reizen.

**Vraag b:** Het A\* algorithm is alleen admissible als geldt dat  $h(n)$  kleiner of gelijk is aan  $h^*(n)$ . Geef een voorbeeld van een zoekboom waarin deze voorwaarde niet geldt, en waaruit blijkt dat A\* in dat geval inderdaad niet admissible is.

**Vraag c:**

- Beschrijf de relatie tussen beam-search en best-first search
- Met welke methode komt beam-search overeen bij een beam-breedte van 1?
- Met welke methode komt beam-search overeen bij een oneindig grote beam-breedte?

## Vraag 2: Zoekmethoden voor 2 spelers



**Vraag a:** Teken bovenstaande zoekboom over, en gebruik het minimax algoritme om te berekenen welke zet de MAX speler zal kiezen boven in de boom. Laat duidelijk blijken hoe je aan dit antwoord komt.

**Vraag b:** Als vraag a, maar nu door alpha-beta-pruning als algoritme te gebruiken. Laat duidelijk blijken hoeveel besparing het alpha-beta-pruning algoritme oplevert in vergelijking met het minimax algoritme uit vraag a.

## Vraag 3: Productieregel systemen & Kennissystemen

**Vraag a:** Beschrijf de reken-cyclus van een productieregel systeem. Vergeet niet te vermelden wanneer de reken-cyclus stopt.

**Vraag b:** Beschrijf hoe we de werking van een productieregel systeem kunnen zien als een zoekboom. Vermeld hoe we de volgende elementen van een zoekboom kunnen herkennen in een productieregel systeem:

- Knopen van de zoekboom
- Takken van de boom
- Diepte van de zoekboom
- Branching-factor van de zoekboom

**Vraag c:** Leg uit wat het belangrijkste voordeel is van de scheiding van kennis en inferentie in de architectuur van kennissystemen, en noem een toepassing waarin dit geïllustreerd wordt.

## Vraag 4: Redeneren met onzekerheid

Gegeven zijn de volgende regels, feiten en zekerheidsfactoren:

- if A and not B then C (0.9)
- if C or D then E (0.75)
- if F then A (0.6)
- if G then A (-0.1)
- F (0.9)
- B (-0.8)
- G (0.7)
- D (0.0)

**Vraag a:** Bereken de zekerheidsfactor van E.

**Vraag b:** Noem minstens 2 nadelen die kleven aan het gebruik van zekerheids factoren.

**Vraag c:** Noem minstens 3 oorzaken van onzekerheid die ervoor zorgen dat we in bijna elke

toepassing van kennissystemen met onzekerheid te maken krijgen.