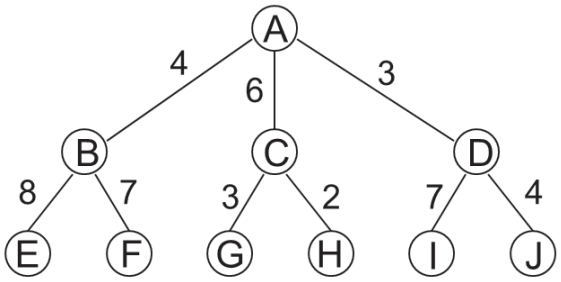
Naam: **Studentnr.:**

Open vragen deeltentamen 1, INFOB2KI 2015-2016.

Deel II

Opgave 1 (10 ptn.) Beschouw onderstaande zoekboom, met daarin tevens weergegeven de *stapkosten*. Stel deze boom is opgebouwd tijdens het toepassen van Uniform Cost Search (UCS).

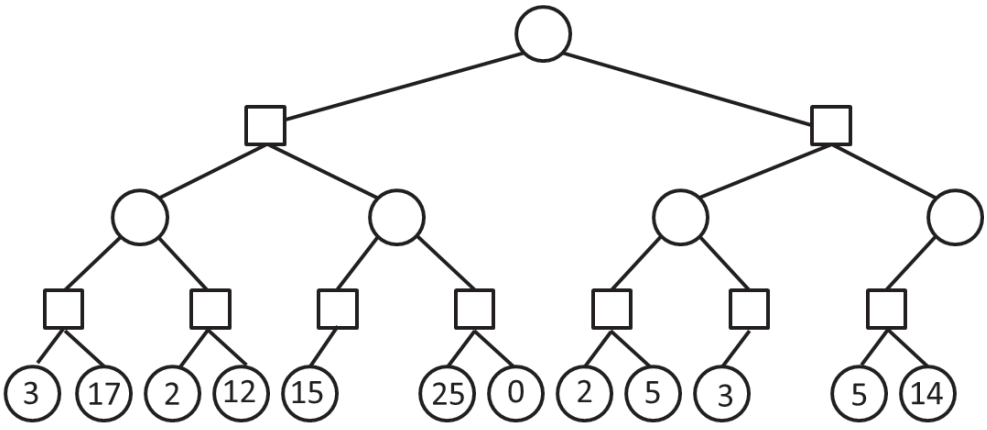


Geef de lijst van alle knopen in de volgorde waarin UCS ze uitklapt.

A, D, B, C, J, H, G, I, F, E

Opgave 2 (10 ptn.) Beschouw onderstaande *game tree* waarin cirkels je eigen zet representeren en vierkanten die van je tegenstander. Pas ***Minimax*** toe; ga er hierbij vanuit dat steeds links vóór rechts gaat. Geef duidelijk in de figuur aan:

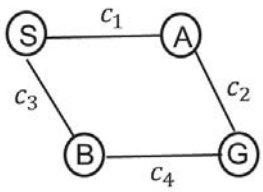
1) de waarde van de evaluatiefunctie die uiteindelijk in de wortel (bovenste knoop) terechtkomt;
2) wat een optimale strategie is.



- Strategie: denk aan het LM/BM voorbeeld: een strategie bestaat uit je eigen zetten _en_ de zetten die je tegenstander _voor_ jouw zetten doet. Een optimale strategy geeft dus jouw beste zet voor elke mogelijke zet van de tegenstander, op elk niveau van de boom.
- Ook toegestaan mits beanrgumenteerd: minimax gaat uit van een optimale tegenspeler. De strategie die behoort bij dit optimale gedrag (en bij de waarde die voor de wortel wordt berekend) is een compleet pad in de gametree; er zijn in dit geval 2 opties voor zo'n pad..
- de waarde van de evaluatiefunctie in de wortel wordt 3

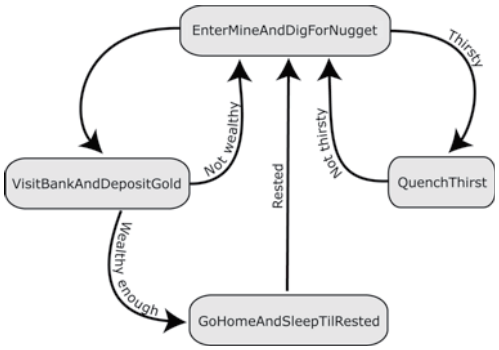
Opgave 3 (10 ptn.) Het 'Greedy-search' zoekalgoritme is in het algemeen niet compleet en daardoor ook niet optimaal. Ook in situaties waarin het algoritme wel compleet is, hoeft het niet optimaal te zijn, zelfs niet als je de *echte* kosten h^* als heuristiek gebruikt.

Beschouw onderstaande routekaart. Geef een voorbeeld-invulling van de (positieve) stapkosten c_i waarbij Greedy-search met heuristiek $h(n) = h^*(n)$ **niet** de kortste route van S naar G geeft.



Let op dat Greedy search niet Greedy kiest op stapkosten, maar op de heuristiek. De heuristieken bij Best First Search (waar Greedy Search een variant op is) geven _altijd_ een inschatting van de *nog te maken* kosten en niet van de gemaakte kosten.
Vergeet niet je antwoord toe te lichten: een invulling van getallen alleen onvoldoende argument om te laten zien dat Greedy search niet de kortste route geeft.

Opgave 4 (10 ptn.) Bob werkt in een goudmijn in het stadje Desert Dirt. In de stad is een bank waar Bob gevonden goudklompjes kan brengen, en een *saloon* om zijn dorst te lessen. Thuis kan hij uitrusten en lekker slapen. Waar Bob heen gaat en wat hij daar doet wordt bepaald door *dorst*, *moeheid* en de *hoeveelheid gevonden goud*. Dit alles is weergegeven in onderstaande *Finite State Machine*.



Bob moet zo nu en dan ook naar het toilet. Geef aan hoe je bovenstaande FSM kan aanpassen zodat Bob naar de wc kan. Geef tevens een voordeel *of* een nadeel van jouw benadering.

Hint: dit betreft een interrupt die in elke state moet kunnen plaatsvinden

Opgave 5 (15 ptn.) Beschouw een Markov beslisproces (MDP) dat het volgende spel modelleert: een speler staat onder aan een trap ('state' 1) en moet boven zien te komen ('state' 4, een 'terminal' state). Er zijn twee tussenliggende treden waarop de speler kan staan ('states' 2 en 3). Na aankomst in eindtoestand 4 krijgt de speler een ***extra*** beloning van ***10***. Bij iedere transitie (toestandsovergang) krijgt de speler een positieve of negatieve beloning op basis van de gekozen actie. Een speler heeft in elke toestand 2 mogelijke acties: *Klim (K)* naar de volgende trede, of *Speel (S)* een minigame die hij kan winnen, gelijkspelen, of verliezen. Bij winst mag hij een trede overslaan; bij gelijkspel blijft hij op de huidige trede en bij verlies moet hij, indien mogelijk, een trede terug. De transitiekansen $T(s,a,s')$ en beloningen $R(s,a,s')$ zijn als volgt (ongespecificeerd is nul):

	$(1,K,2)$	$(2,K,3)$	$(3,K,4)$
T	1	1	1
R	2	2	2

	$(4,\text{exit},\perp)$
T	1
R	10

	$(1,S,1)$	$(1,S,3)$	$(2,S,1)$	$(2,S,2)$	$(2,S,4)$	$(3,S,2)$	$(3,S,3)$	$(3,S,4)$
T	0.4	0.6	0.1	0.3	0.6	0.1	0.3	0.6
R	-1	5	-1	-1	5	-1	-1	5

De optimale waarden $V^*(s)$ worden met behulp van Value Iteration uit de Bellman vergelijking met $\gamma = 1$ bepaald:

s	1	2	3	4
$V^*(s)$	19.0	16.7	14.7	10.0

$$V^*(s) = \max_a \sum_{s'} T(s,a,s')\Big(R(s,a,s') + \gamma V^*(s')\Big)$$

Bereken de optimale policy π^* voor de speler.

Hierbij wordt alleen om een optimal policy gevraagd. In zo'n klein voorbeeld is het eenvoudig in te zien wat de optimal policy is zonder de berekeningen uit te voeren. De opgave zegt expliciet dat je de optimale policy moet *berekenen* dus moet duidelijk zijn dat je inderdaad gerekend hebt: kan je de Bellman equation toe passen? Let op dat V* in de functie de waarden uit de vorige iteratie zijn, en niet uit de huidige (dus allemaal waarden uit de gegeven tabel!).
V* is al gegeven, dus je moet 'm 1 stap doorrekenen om te bepalen bij welke acties de gegeven waarden horen. Let op dat een policy voor *elke* state een actie specificeert, dus ook voor state 4!
Optimal policy volgt uit de volgende waarden (bereken ze zelf):
1: Klim -> 18,7; Speel -> 19
2: Klim -> 16.7; Speel -> 15.5
3: Klim -> 12; Speel -> 14.7
4: exit -> 10 (enige mogelijkheid)

Opgave 6 (15 ptn.) Beschouw een robot die zich met ***reinforcement learning*** leert bewegen door een wereld. De robot kent 7 toestanden ('states') $A \ldots G$, twee acties: L (links) en R (rechts) en beweegt via een vaste policy π. De robot heeft 2 epochs/episodes getraind en onderstaande transities $< s, a, r, s' >$ ervaren:

Epoch 1: $< A, L, 5, B >, < B, L, 4, D >, < D, R, -5, C >, < C, R, 2, G >$
Epoch 2: $< A, L, 5, B >, < B, L, -5, E >, < E, L, -2, G >, < G, R, 4, F >$

We beschouwen nu zowel Direct Evaluation voor het bepalen van de waarden $V^\pi(s)$ voor elke toestand s, als Q-learning voor het bepalen van $Q(s, a)$-waarden. Geef in onderstaande tabellen:

I. de uitkomsten $V^\pi(s)$ voor $\gamma = 1$
II. de uitkomsten $V^\pi(s)$ voor $\gamma = 0.5$
III. een vinkje voor die $Q(s, a)$-waarden die tijdens bovenstaande training ge-update worden; geef ook duidelijk de bijbehorende acties aan.

Gebruik de ruimte in de tabellen voor zover nodig; geef wel duidelijk aan wat er in elke rij staat.

	A	*B*	*C*	*D*	*E*	*F*	*G*
I	4	-1	2	-3	2	x	4

	A	*B*	*C*	*D*	*E*	*F*	*G*
II	4.25	-1.5	2	-4	0	x	4

	A	*B*	*C*	*D*	*E*	*F*	*G*
III L	u	u			u		
R			u	u			u

- voor DE (vraag I en II):
  - let op: dit betreft "reward-to-go", gemiddeld over het aantal keren dat een state bezocht is (niet perse gelijk aan aantal epochs). Hoe verder weg de rewards, hoe meer verdisconteerd.Waarde wordt eenmalig bepaald, nadat alle epochs doorlopen zijn.

- voor Q-tabel (vraag III):
  - het verschil tussen V-waarden en Q-waarden is dat V-waarden per toestand gegeven worden en Q-waarden voor ieder toestand-actie paar. Er moest dus wel duidelijk aangegeven worden voor welke _actie_ dan de Q-waarde bij een toestand werd ge-update; update gebeurt na ieder bezoek..