1) **Co dělal AI systém Shakey?**
    - general purpose robot, Lisp jazyk, A* algoritmus, plánování STRIPS, hrozně se třásl

2) **Uspořádejte následující AI systémy od nejstaršího po nejnovější:**
   DeepBlue
   AlphaGo
   Shakey
   AlphaStar

    - Shakey, DeepBlue, AlphaGo, AlphaStar

3) **Kde se poprvé objevilo slovní spojení “artificial intelligence”?**

    A. v disertační práci Allana Turinga
    B. v pracích Leonarda da Vinci
    C. v návrhu konání workshopu v Dartmouth

4) **Co byl systém MYCIN?**

    - Odpověď: Byla to umělá inteligence, která doporučovala léky (antibiotika). Byl napsán v Lispu. Měl přesnost 65%, což bylo ale lepší, než člověk.

5) **Na jakém problému byl poprvé předveden algoritmus DQN?**

    A. Go
    B. Jeopady!
    C. Šachy
    D. Poker
    E. Atari hry

6) **Popište krátce co znamená pojem “AI Winter” a proč k němu dochází.**
    - Úpadek zájmu o další vývoj umělé inteligence v daném období. Dochází k němu například z důvodu nepřesvědčivých výsledků předcházejícího výzkumu, či kvůli tomu, že v dané době pro takovéhle algoritmy ještě neexistoval dostatečný hardware.

7) **Co dělal AI systém ELIZA?**
    - Chatbot, psychoterapeut, na základě vstupu tvořil jednoduché otázky a věty, případně defaultní zprávy

8) **Co je Turingův test a k čemu slouží?**
    - Test co má za úkol rozlišit počítač od člověka. V první verzi - šachy, dále úprava -> porota má rozhodnout zda jim na otázky odpovídá člověk či stroj (stroj a člověk jsou v oddělených místnostech, kam porota nevidí).
    - Rozdíl weak / strong AI:
      weak AI jen simuluje schopnost rozumět
      strong AI skutečně rozumí

9) **Kterou loterii by zvolil racionální agent?**

A. 5% šance získat $40
B. 100% šance získat $2
C. 1% šance získat $300
D. 50% šance získat $8

(racionální agent vybírá možnost, která maximalizuje užitek)

10) **Kterou loterii by zvolil racionální agent?**

A. 100% šance získat $8
B. 30% šance získat $20
C. 10% šance získat $40
D. 2% šance získat $400

11) **Kterou z následujících možností by zvolil racionální agent?**

A. Dolar za každý hod férovou mincí při kterém padne panna, dokud nepadne první orel
B. Počet dolarů daný jedním hodem standardní šestistěnnou kostkou
C. 3 dolary a 1 cent

12) **Reprezentujte následující pravděpodobnostní distribuci jako Bayesovskou síť**

	toothache		*¬toothache*	
	catch	*¬catch*	*catch*	*¬catch*
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

Definujte všechny potřebné komponenty. Pro pravděpodobnostní tabulky definujte kterou pravděpodobnostní distribuci reprezentují a můžete vynechat hodnoty, které jsou doplňkem uvedených hodnot do 1.

13) **Kolik položek má sdružená pravděpodobnostní distribuce nad *n* binárními proměnnými?**

A. n^3
B. 2^n
C. n!
D. n
E. n^2

14) **Které z následujících tvrzení jsou vždy pravdivé?**
(pozn.: tady prý není správná odpověď)

A. P(A, B) = P(A) * P(B)
B. P(A|B) = P(B|A)
C. P(A|B) = P(B|A) * P(A)

15) Které z následujících tvrzení jsou vždy pravdivé?

A. P(A, B) = P(B|A) * P(A)
B. P(A, B) = P(A) * P(B)
C. P(A|B) = P(B|A)

16) Označte všechny operace, které musíme provést na STRIPS akci *a*, abychom vytvořili její relaxovanou variantu a^+.

A. Smažeme její cenu (cost)
B. Smažeme pozitivní efekty (add effects)
C. Smažeme její negativní efekty (delete effects)
D. Smažeme její předpoklady (preconditions)

17) Které z následujících tvrzení o automatickém plánování jsou pravdivé?

A. Plánování vyžaduje úplný formální popis problému.
B. Jeden z algoritmů běžně používaných v plánování je A*.
C. Korektní plán je typicky nalezen v čase polynomiálním ve velikosti popisu plánovacího problému.

18) Uvažujme POMDP, kde:
- **Počet stavů = 15**
- **Počet možných pozorování = 4**
- **Počet akcí = 2**

Jaký je počet alfa-vektorů po 2 krocích plné iterace hodnot (value iteration)?

A. $4 * 4^2 = 64$
B. 15 * 4 = 60
C. $2 * 2^4 = 32$
D. 4

Vzorec ze cvičení:
|V_0| = |A|
|V_t+1| = |A| * |V_t| ^ |O|

19) Uvažujme POMDP, kde:
- **Počet stavů = 10**
- **Počet možných pozorování = 2**
- **Počet akcí = 4**

Jaký je počet alfa-vektorů po 2 krocích plné iterace hodnot (value iteration)?

A. $10 * 4 = 40$
B. 4
C. $4 * 4^2 = 64$
D. $4 * 2^4 = 64$

20) Uvažujme POMDP, kde:

- **Počet stavů = 5**
- **Počet možných pozorování = 3**
- **Počet akcí = 2**

Jaký je počet alfa-vektorů po 2 krocích plné iterace hodnot (value iteration)?

A. $2 * 2^3 = 16$
B. $2 * 2^5 = 64$
C. $5 * 2 * 3 = 30$
D. $2 * 5^3 = 250$

21) Co je alfa-vektor?

A. Lineární funkce vyjadřující očekávanou pravděpodobnost pro zvolený stav světa a danou strategii v závislosti na měnícím se beliefu agenta.
B. Lineární funkce vyjadřující očekávaný zisk pro danou strategii agenta v závislosti na měnícím se beliefu agenta.
C. Lineární funkce vyjadřující očekávaný zisk pro iniciální belief agenta v závislosti na měnící se strategii

22) Vyznačte ta tvrzení o POMDP, která jsou pravdivá

A. Belief je pravděpodobnostní distribuce nad stavy.
B. alfa-vektor odpovídá očekávané hodnotě vždy jedné akce agenta.
C. Hodnota v iniciálním beliefu se v případě algoritmu iterace hodnot (value iteration) blíží optimální hodnotě shora.

23) Vyznačte ta tvrzení o POMDP, která jsou pravdivá

A. Belief je pravděpodobnostní distribuce nad akcemi.
B. Množina všech dosažitelných pravděpodobnostních distribucí, které odpovídají možným belief stavům v problému, nezávisí na strategii agenta.
C. POMDP odpovídají MDP se spojitým stavovým prostorem.

24) Předpokládejme, že agent v multi-armed bandit problému pozoruje následující sekvenci odměn:

A	2	1	1.5			0		
B				1.5			0	1.5
C					1.5			

Jaká je pravděpodobnost výběru následující akce (a) greedy agentem, (b) epsilon-greedy agentem s explorací 0.1 a (c) UCB agentem s exploračním parametrem nastaveným na 4. Zapište všechny tři pravděpodobnostní distribuce.

A = 4.5/4 = 1.125
B = 3/3 = 1
C = 1.5/1 = 1.5
a) p(A) = p(B) = 0, p(C) = 1
~~b) p(A) = p(B) = 0.05, p(C) = 0.9~~
c) nevím přesně, jak to přepsat do pravděpodobnosti, ale hodnoty jsou
A = 1.125 + 4sqrt(log9/4)
B = 1 + 4sqrt(log9/3)
C = 1.5 + 4sqrt(log9/1)
z čehož největší je C, tedy UCB agent by zvolil C

a)
vybere vždy C, protože je tam největší avg gain
P(C) = 1
P(A) = 0
P(B) = 0
b)
vybere C s pravděpodobností 0.9, a náhodnou akci (A,B,C)
P(C) = 0.933
P(A) = 0.033
P(B) = 0.033

25) Předpokládejme, že agent v multi-armed bandit problému pozoruje následující sekvenci odměn:

A	1		0	0.5	2	1.5		
B		0.5						1.5
C							0	

Jaká je pravděpodobnost výběru následující akce (a) greedy agentem, (b) epsilon-greedy agentem s explorací 0.1 a (c) UCB agentem s exploračním parametrem nastaveným na 4. Zapište všechny tři pravděpodobnostní distribuce.

- a) P(A) = P(B) = 1/2, P(C) = 0
- b) P(A) = 0,4 + 0,033, P(B) = 0,4 + 0,033, P(C) = 0,033
- c) A = 5 + 4*sqrt(log9/5), B = 2 + 4*sqrt(log9/2), C = 0 + 4*sqrt(log(9/1) -> vybere C, protože je největší

26) Předpokládejme, že agent v multi-armed bandit problému pozoruje následující sekvenci odměn:

A	2				2	1.5		0.5
B			2	2			2	
C		0.5						

Jaká je pravděpodobnost výběru následující akce (a) greedy agentem, (b) epsilon-greedy agentem s explorací 0.1 a (c) UCB agentem s exploračním parametrem nastaveným na 4. Zapište všechny tři pravděpodobnostní distribuce.

27) Vyznačte které z následujících tvrzení jsou pravdivé

A. Prohledávání do šířky vždy nalezne řešení, pokud toto existuje.
B. Každá konzistentní heuristika je také přípustní heuristikou.

C. Počet nesprávně seřazených čísel (tj. menší číslo je za větším) je přípustnou heuristikou pro 8-puzzle problém.
D. Prohledávání do hloubky má vyšší paměťové nároky než prohledávání do šířky.
E. Předpokládejme, že všechny ceny (costs) akcí jsou celá čísla a jsou striktně pozitivní. Funkce h(x) = 1 pro všechny stavy x, kromě cíle, je přípustnou funkcí.

28) Vyznačte které z následujících tvrzení jsou pravdivé

A. Prohledávání do hloubky má vyšší paměťové nároky, než prohledávání do šířky.
B. Prohledávání do hloubky najde vždy optimální řešení i v nekonečném stavovém prostoru.
C. Každá konzistentní heuristika je také přípustní heuristikou.
D. Předpokládejme, že všechny ceny (costs) akcí jsou celá čísla a jsou striktně pozitivní. Funkce h(x) = 1 pro všechny stavy x, kromě cíle, je přípustnou funkcí.
E. Počet nesprávně umístěných čísel je přípustnou heuristikou pro 8-puzzle problém.

29) Vyznačte které z následujících tvrzení jsou pravdivé

A. Prohledávání do šířky (BFS) vždy expanduje méně uzlů než je prohledávání do hloubky (DFS)
B. A* musí používat konzistentní heuristiku, aby našel optimální řešení
C. Za Předpokladu, že existuje řešení a všechny akce mají striktně pozitivní cenu (cost), iterativní prohledávání do hloubky IDA* s h(x)=0 pro všechny stavy vždy najde optimální řešení i v nekonečném stavovém prostoru.
D. Počet nesprávně umístěných čísel je přípustnou heuristikou pro 8-puzzle problém
E. Manhattanská vzdálenost je přípustná heuristika pro plánování robota v bludišti (na mřížce) pokud se robot může pohybovat všemi 8 směry.

30) Vyznačte které z následujících tvrzení jsou pravdivé (dopsat)

A. Počet nesprávně seřazených čísel (tj. menší číslo je za větším) je přípustnou heuristikou pro 8-puzzle problém.
B. Každá přípustní heuristika je také konzistentní heuristikou.
C. Obousměrné (bidirectional) informované prohledávání vždy expanduje méně uzlů než jednosměrné prohledávání
D. Počet nesprávně umístěných čísel je přípustnou heuristikou pro 8-puzzle problém
E. Za předpokladu, že existuje řešení a všechny ceny (costs) za akce se rovnají 2, prohledávání do šířky vždy najde optimální řešení i v nekonečném stavovém prostoru

31) Mějme problém s vysáváním místnosti reprezentované jako obdélník s X x Y bla bla bla dopsat později

32) Mějme problém s vysáváním místnosti reprezentované jako obdélník s X x Y bla bla bla dopsat později

33) Uvažte následující CSP

- Proměnné: x_1, x_2, x_3, x_4
- Domény pro každou proměnnou: {0, 1, 2, 3}
- Omezení:

$x_1 \neq x_2$
$x_1 = x_3 + 1$
$x_2 < x_3$
$x_4 = 2 * x_2$

Zapište domény pro každou proměnnou po provedení kompletního AC-3 algoritmu:

$x_1 =$
$x_2 =$
$x_3 =$
$x_4 =$

34) Uvažte následující CSP

- Proměnné: x_1, x_2, x_3, x_4
- Domény pro každou proměnnou: {0, 1, 2, 3}
- Omezení:

$x_1 > x_2$
$x_2 > x_3$
$x_4 = 1 + x_2$

Zapište domény pro každou proměnnou po provedení kompletního AC-3 algoritmu:

$x_1 =$
$x_2 =$
$x_3 =$
$x_4 =$

35) Uvažte následující CSP

- Proměnné: x_1, x_2, x_3, x_4
- Domény pro každou proměnnou: {0, 1, 2, 3}
- Omezení:

$x_1 = x_3 + 1$
$x_2 < x_3$
$x_3 = x_4$
$x_4 = 2 * x_2$

Zapište domény pro každou proměnnou po provedení kompletního AC-3 algoritmu:

$x_1 =$
$x_2 =$
$x_3 =$
$x_4 =$

36) Uvažte následující CSP

- Proměnné: x_1, x_2, x_3, x_4
- Domény pro každou proměnnou: {0, 1, 2, 3}
- Omezení:

$x_1 < x_2$

$x_2 < x_3$
$x_4 = 2 * x_2$

Zapište domény pro každou proměnnou po provedení kompletního AC-3 algoritmu:

x_1 =
x_2 =
x_3 =
x_4 =

37) Algoritmus CSP

A. nemusí vždy najít řešení
B. hledá nejkratší cestu ze startu do cíle
C. hledá přípustné přiřazení hodnot proměnných z jejich domén

38) Která z následujících tvrzení jsou pravdivá?

A. Negamax prohledá stejné uzly herního stromu jako minimax.
B. Alpha-beta může prohledat méně různých uzlů herního stromu jako Negascout.
C. Negascout navštíví uzly v herním stromě nejvýše dvakrát.

39) Která z následujících tvrzení jsou pravdivá?

A. UCT může prohledat i uzly, které alpha-beta nebude prohledávat.
B. Negascout vždy navštíví méně uzlů než minimax.
C. Negamax navštíví každý uzel v herním stromě nejvýše jednou.

40) Vyznačte, která z následujících tvrzení jsou pravdivé

A. Algoritmus Monte Carlo Tree Search staví vybalancovaný (z hlediska teorie grafů) herní strom
B. Negascout může navštívit některé uzly v herním stromě opakovaně
C. Alpha-beta prořezávání vždy navštíví méně uzlů herního stromu než minimax

41) Je následujících tvrzení pravdivé?
Pro řešení CSP můžeme využít i algoritmus A*.

A. Ano
B. Ne

42) Čemu odpovídá hrana v CSP prohledávacím stromě (search tree)?

A. Proměnné CSP problému.
B. Hodnotě přiřazené proměnné (uzlu, ze kterého hrana vychází)
C. Množině dostupných hodnot pro proměnnou reprezentovanou uzlem, ze kterého hrana vychází

43) Čemu odpovídá uzel v CSP prohledávacím stromě (search tree)?

A. Hodnotě přiřazené proměnné (uzlu, ze kterého hrana vychází)
B. Proměnné CSP problému.
C. Množině nepřiřazených proměnných

44) Které z následujících tvrzení o algoritmu AlphaGo jsou pravdivé?

A. Algoritmus AlphaGo nebyl zatím v hraní Go překonán žádným člověkem ani jiným AI systémem.
B. Algoritmus používal tři různé policy a jedna z nich aproximuje lidskou hru.
C. Algoritmus porazil profesionálního hráče ve hře Go.

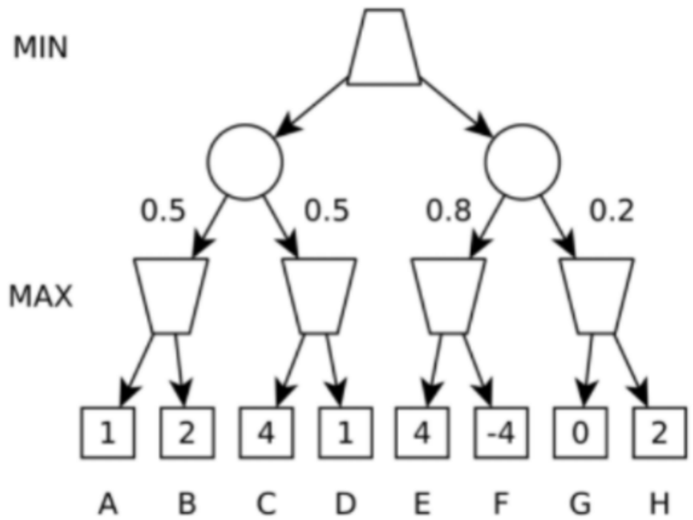
45) Je následujících tvrzení pravdivé?
AC-3 algoritmus je spuštěn v rámci řešení CSP prohledáváním nejvýše jednou.

A. Ano
B. Ne

46) Uvažme problém CSP s dvěmi proměnnými x_1, x_2, obě s doménou {a, b}, a podmínkou $x_1 \neq x_2$. Algoritmus AC-3 v tomto případě:

A. Zredukuje domény některých proměnných
B. Zjistí, že je problém neřešitelný
C. Nic neudělá

47) Zvažte hru se dvěma hráči s nulovým součtem zobrazenou na obrázku níže.



Tato hra obsahuje stochastické události modelované náhodnými uzly (kruhy ve stromu hry) s pravděpodobnostmi napsanými nad akcemi vedoucími z těchto uzlů.

1. **Spočtěte očekávanou hodnotu hry, pokud hráči hrají podle optimálních strategií.**

2. **Představte si modifikaci alfa-beta prořezávání pro hry s náhodnými uzly. Předpokládejme, že akce jsou vyhodnocovány zleva doprava a předpokládáme, že hodnoty užitku v listech jsou přiřazovány z intervalu [-inf, inf]. Které listy může tento modifikovaný alfa-beta algoritmus uritě přeskočit?**
3. **Jak by se situace změnila, kdybychom věděli, že hodnoty užitku v listech jsou přiřazovány z intervalu [-4, 4]? Které listy může tento modifikovaný alfa-beta algoritmus určitě přeskočit?**

48) Uvažujme standardní alfa-beta prořezávání s upraveným počátečním alfa-beta intervalem nastaveným na [w, x]. Algoritmus vrátí hodnotu y < w. Které z následujících tvrzení je pravdivé?

A. Přesná hodnota hry není známa, ale je menší nebo rovna hodnotě w.
B. Hodnota hry je známá a je rovna hodnotě w.
C. Přesná hodnota hry není známa, ale je menší nebo rovna hodnotě y
D. Hodnota hry je známá a je rovna hodnotě y.

49) Co musíme nezbytně udělat pro transformaci epizodické úlohy do ekvivalentní pokračující úlohy ve zpětnovazebném učení?

A. Nastavit diskontní míru gamma E (0, 1)
B. Nahradit terminální stavy absorbujícími stavy (i.e., self-loops)
C. Přeškálovat odměny tak, aby byly menší než jedna

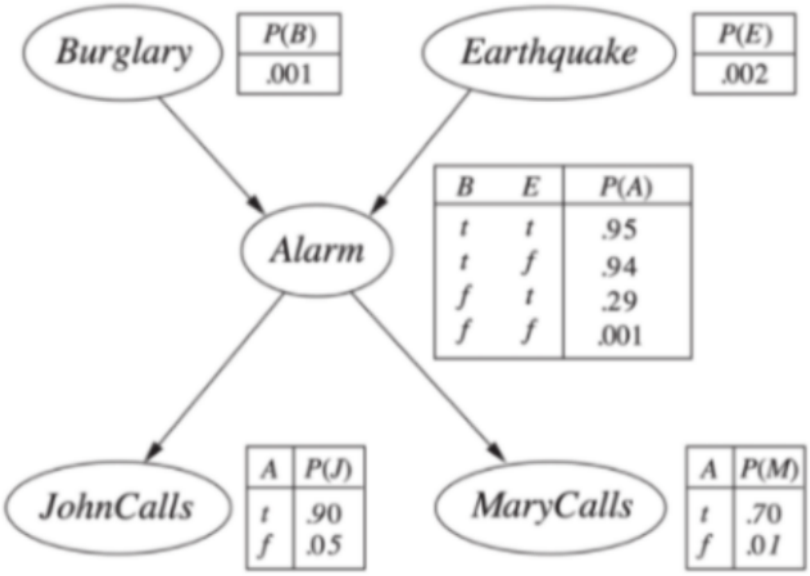
50) Předpokládejme MDP s A akcemi a S stavy. Které z následujících tvrzení jsou pravda?

A. Časová složitost u jednoho kroku Q-learning je $O(S^2 * A)$
B. Časová složitost u jednoho kroku Q-learning je $O(A)$
C. Prostorová složitost u tabulkového Q-learning je $O(S)$
D. Prostorová složitost u tabulkového Q-learning je $O(S^2 * A)$
E. Prostorová složitost u tabulkového Q-learning je $O(S * A)$
F. Časová složitost u jednoho kroku Q-learning je $O(S * A)$
G. Prostorová složitost u tabulkového Q-learning je $O(S * A^2)$
H. Časová složitost u jednoho kroku Q-learning je $O(S * A^2)$

51) Jaké vstupy musí mít agent ve zpětnovazebním učení, aby mohl najít optimální strategii (optimal policy)?

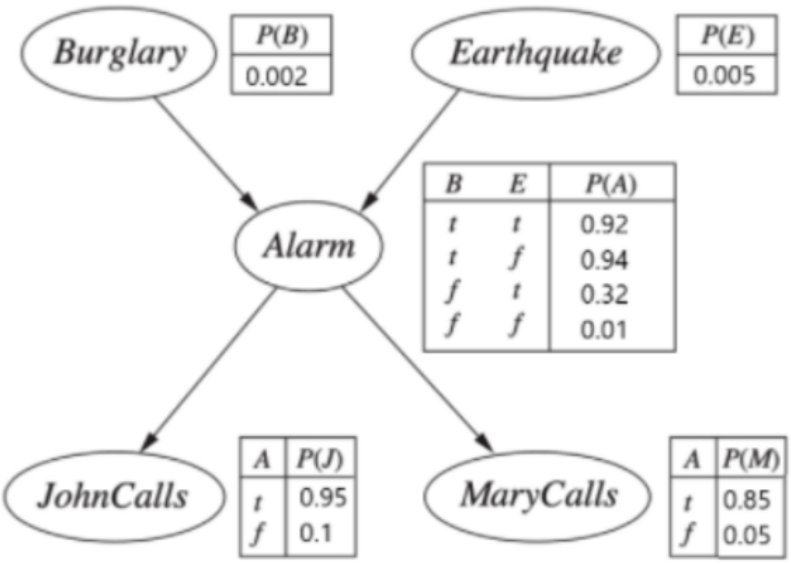
A. Množina všech možných stavů
B. Přesný popis odměnových dynamik
C. Funkce hodnot (Value function)
D. Množina všech dostupných akcí
E. Simulátor prostředí
F. Počet možných stavů
G. Konečná délka horizontu
H. Přesný popis přechodových dynamik

52) Uvažujme následující Bayesovskou síť



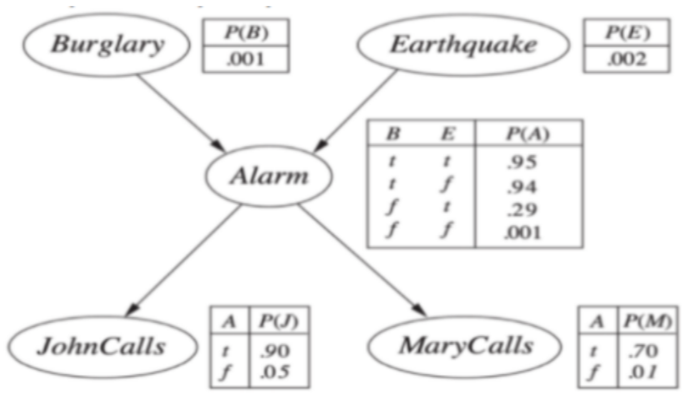
Jaká je pravděpodobnost, že oba John i Mary zavolají, když víme, že nastalo zemětřesení (Earthquake). Popište postup, jak jste k výsledku došli.

53) Uvažujme následující Bayesovskou síť



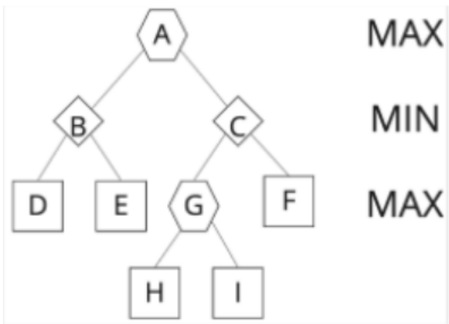
Jaká je pravděpodobnost, že alespoň jeden ze sousedů (John nebo Mary) zavolá, když víme, že nastalo vloupání (Burglary). Popište postup, jak jste k výsledku došli.

54) Uvažujme následující Bayesovskou síť



Jaká je pravděpodobnost, že alespoň jeden ze sousedů (John nebo Mary) zavolá, když víme, že nastalo vloupání (Burglary). Popište postup, jak jste k výsledku došli.

55) Uvažujte dvouhráčovou hru na obrázku



Napište utility hodnoty pro listy stromu (uzly D, E, H, I, F), tak aby algoritmus Alpha-Beta nic neprořezal a algoritmus Negascout prořezal některé z uzlů

- 5 6 7 **8** 4
- 4 4 7 **-1** 3

56) Vypište tři metody, založené na různém principu, které podporují exploraci při zpětnovazebném učení

- Odpověď:

57) Uvažujme problém nalezení nejrychlejšího spojení v MHD za předpokladu, že můžete použít různé způsoby dopravy (tramvaj, autobus, pěší přesun mezi zastávky do vzdálenosti 10 minut, etc.)

1. **Navrhněte formální reprezentaci, kterou by bylo možné použít pro prohledávání prostoru.**
2. **Navrhněte netriviální přípustnou heuristiku a zdůvodněte proč je přípustná**

- Odpověď:

58) Uvažujme problém plánování pohybu robotické ruky v prostoru (např. pro manipulaci s předměty) s cílem vykonat co možná nejefektivnější pohyb bez nechtěné kolize s předmětem.

1. **Navrhněte formální reprezentaci, kterou by bylo možné použít pro prohledávání prostoru.**

2. **Navrhněte netriviální přípustnou heuristiku a zdůvodněte proč je přípustná**

- Odpověď:
- 1. Může to být MDP s reprezentací polohy a ostatních objektů {α1, α2, α3, armLength, …}
- 2. Heuristiku můžeme dělat podle pythagorové věty

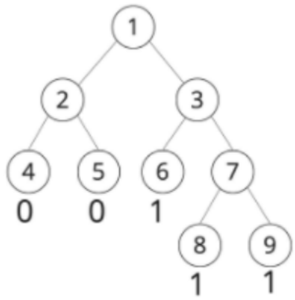
59) Který selekční algoritmus by jste použili v Monte Carlo prohledávání stromu s velkým (případně nekonečným) počtem akcí?

- Odpověď:

60) Setřiďte fáze Monte Carlo prohledávání stromu podle počtu různých stavů hry, které jsou během každé fáze typicky navštíveny od nejmenšího.

- Odpověď:

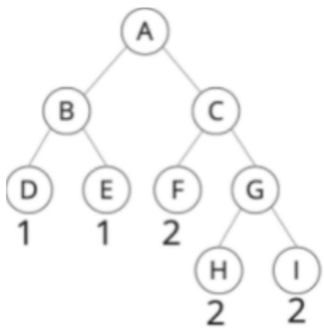
61) Uvažujte následující jednohráčovou hru (čísla pod listy značí odměnu hráče v daném stavu):



Kolik iterací musí algoritmus UCT s explorací 2 vykonat, aby bylo zaručené, že navštíví stav 4? Napište konkrétní číslo a pak jeho zdůvodnění.

- Odpovedʼ:

62) Uvažujte následující jednohráčovou hru (čísla pod listy značí odměnu hráče v daném stavu):



Uvažujme algoritmus UCT s explorační konstantou c = 2 a předpokládejme, že při rovnosti hodnot algoritmus preferuje levou akci (neplatí pro simulaci).

1. **Specifikujte postavenou část stromu po 5 iteracích algoritmu**

2. **Ve které iteraci UCT algoritmu se garantovaně navštíví uzel D? Uveďte pořadí iterace i postup výpočtu**

- Odpověď:

63) CSP prohledávání sématicky odpovídá prohledávání

A. do šířky
B. do hloubky
C. do hloubky s iterativním prodlužováním horizontu (iterative deepening)

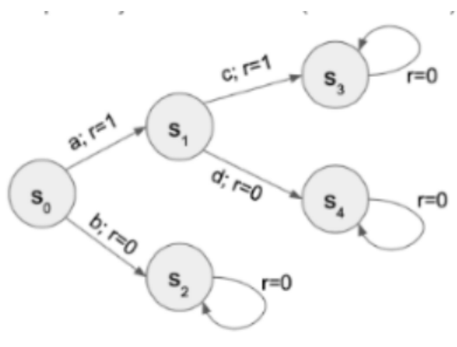
64) Předpokládejme deterministický gridworld, kde se robot může pohybovat jen doleva (L) nebo doprava (R). Pohyb nic nestojí a robot získá odměnu 4 pokud opustí herní plochu. Diskontní míra (discount factor) je 1. Předpokládejme, že iniciální value funkce je:

2	1	0	1	0

(a) jaké jsou hodnoty value funkce po dvou krocích value iteration algoritmu?
(b) jaké strategie by se měl držet agent s value funkcí z bodu (a) ve stavu nejvíce vlevo a proč?
(c) jak byste změnili tuto strategii aby fungovala lépe při praktické aplikaci v robotice?

a. Přidáme krajní stavy iniciální value funkce: 4 | 2 1 0 1 0 | 4
   Po prvním kroku: 4 | 4 2 1 0 4 | 4
   Po druhém kroku: 4 | 4 4 2 4 4 | 4
b. Jít pouze vlevo
c. Přidat negativní reward za pohyb

65) Vypočítejte přesné hodnoty <u>optimálních</u> value funkcí ve všech neabsorbujících stavech a pro všechny akce v následujícím MDP. Předpokládejme diskontní míru (discount factor) 0.5.



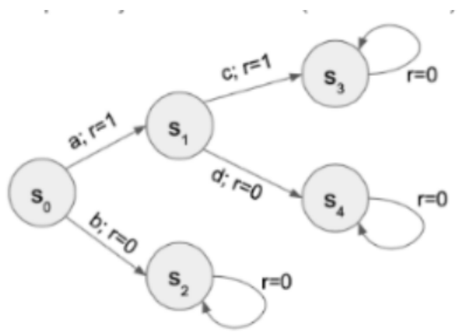
- Odpověď: V(s) = max(R(a) + alfa*V(s'))

  v(s3) = 0
  v(s4) = 0
  v(s2) = 0
  v(s1) = 1

v(s0) = 1.5

v(b) = 0
v(d) = 0
v(c) = 1
v(a) = 1.5

66) (**Pozn.: Pozor, zní podobně, ale tato úloha je jiná od č. 65**) **Vypočítejte přesné hodnoty value funkcí pro neabsorbující stavy a pro akce pro uniformní strategii (policy) v následujícím MDP. Předpokládejme diskontní míru (discount factor) 0.5.**

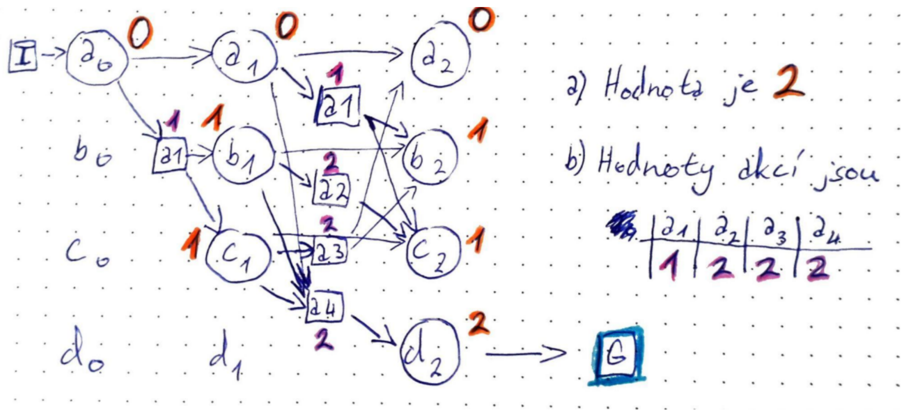


67) **Uvažujme STRIPS problém s predikáty a, b, c, d, iniciálním stavem I = {a}, akcemi**

a_1 =< {a}, {b, c}, ∅ >
a_2 =< {b}, {c}, ∅ >
a_3 =< {c}, {a, b}, ∅ >
a_4 =< {a, b, c}, {d}, ∅ >
a cílovým stavem G = {d}

(a) **Jaká je hodnota heuristiky h_{max} v iniciálním stavu?**
(b) **Jaká je hodnota každé z akcí během výpočtu h_{max} v iniciálním stavu?**

- Odpověď:

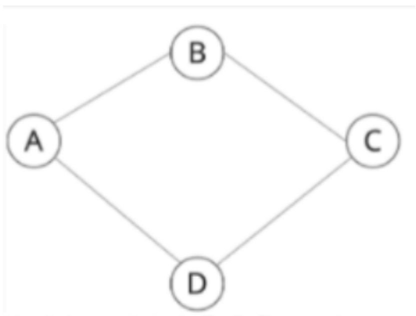


viz https://cw.fel.cvut.cz/old/_media/courses/a4m33pah/03-relaxation.pdf

68) **Uvažujme STRIPS problém s predikáty a, b, c, d, iniciálním stavem I = {a}, akcemi**
a_1 =< {a}, {b}, ∅ >
a_2 =< {b}, {c}, ∅ >
a_3 =< {c}, {a, b}, ∅ >
a_4 =< {a,b,c}, {d}, ∅ >
a cílovým stavem G = {d}

a) **Jaká je hodnota heuristiky h_{max} v iniciálním stavu?**
b) **Jaká je hodnota každé z akcí během výpočtu h_{max} v iniciálním stavu?**

69) **Uvažujme města A, B, C, D propojeny cestami na obrázku. Ve městě A je zásilka. Ve městě C je auto. Auto se může přesunout do sousedního města, naložit nebo vyložit zásilku.**



pomocí formalismu STRIPS popište:
(a) **tuto iniciální situaci**
(b) **akce naložení zásilky, vyložení zásilky a přesunu auta**

- Odpověď:
- a)
- P = {
  car_at_a/b/c/d
  box_at_a/b/c/d
  box_loaded
  adjacent_a_b, adjacent_b_c, adjacent_c_d, adjacent_d_a
  }

  I = {car_at_c, box_at_a }
  G = {}

  b)
  load = < {car_at_x, box_at_x}, {box_loaded}, {box_at_x}>
  unload = < {car_at_x, box_loaded}, {box_at_x}, {box_loaded}>
  move_x_y = < car_at_x, adjacent_x_y}, {car_at_y}, {car_at_x}>
  (Misto x a y by to asi melo bejt pro vsechny a,b,c,d a kombinace)

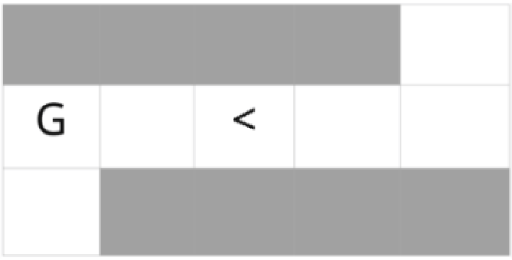
70) Uvažujme následující problém navigace robota v bludišti…

Uvažujme následující problém navigace robota v bludišti, kde robot nezná přesnou polohu a získává pouze pozorování (observation) z 4-okolí, zlato (G) není možné pozorovat. Robot se může dopředu pohybovat v daném směru, nebo se může otočit vpravo nebo vlevo.

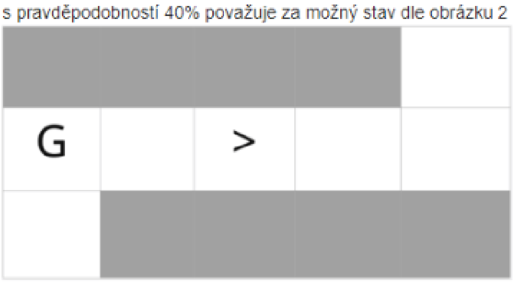
Vykonání každé akce může selhat s pravděpodobností 5% -- v tom případě nedojde ke změně stavu světa.

Za každou akci získá robot negativní ohodnocení -0.1. Za vkročení na políčko se zlatem (G) získá robot ohodnocení 10.

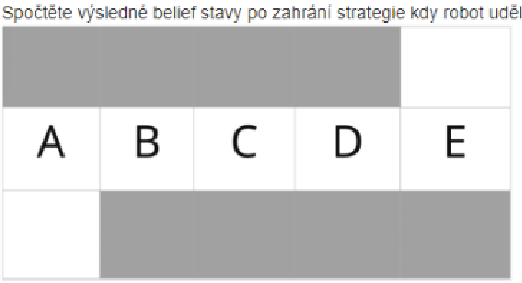
Předpokládejme, že robot má následující belief:
s pravděpodobností 60% považuje za možný stav dle obrázku 1



s pravděpodobností 40% považuje za možný stav dle obrázku 2



1. Spočtěte výsledné belief stavy po zahrání strategie kdy robot udělá 2 kroky vpřed bez ohledu na získané pozorování.



2. Spočtěte očekávanou hodnotu strategie kdy robot udělá 2 kroky vpřed bez ohledu na získané pozorování. Uveďte konkrétní hodnotu i postup výpočtu.