Umetna inteligenca Ime in priimek: _____

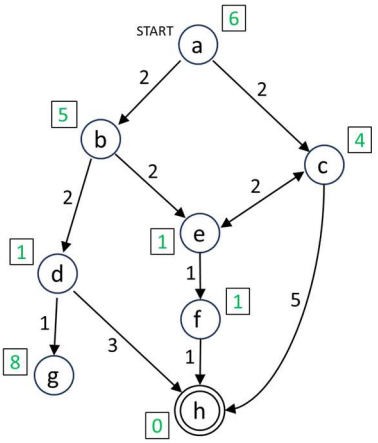
Kolokvij 1/2, 2024/25 Vpisna številka: _____

Odgovarjajte kratko in jedrnato, točno na zastavljena vprašanja. Vse odgovore pišite na črto pod vprašanji in **izključno na ta list**, ki ga edinega oddate na koncu! Čas pisanja: 45 minut.

---

(1)

Na sliki je podan prostor stanj. Začetno vozlišče je **a**, ciljno vozlišče je **h**. Nasledniki nekega vozlišča se *vedno* generirajo po abecednem vrstnem redu. Poleg vozlišč je v okvirčku z zeleno barvo podana še hevristična ocena danega vozlišča. Bodite pozorni, da je povezava **c-e** dvosmerna, cena povezave je v obe smeri enaka! Vprašanja spodaj se navezujejo na ta primer, če ni navedeno drugače.



(a) Katero rešitev vrne iskanje v globino?

____**a – b – d – h**_____

(b) Katera vozlišča (tudi morebitne ponovitve) in v kakšnem vrstnem redu so na koncu iskanja v širino v vrsti za razvijanje (npr. če bi se algoritem nadaljeval)?

_____ **g h c f c f**_____

(c) Kakšen je vrstni red razvitih vozlišč pri iterativnem poglabljanju (vključno s ponovitvami, seveda)?

_____ **a | a b c | a b d e c e h** _____

(d) Koliko dodatnega dela (razvitih vozlišč) bi *v splošnem* opravilo iterativno poglabljanje v primerjavi z iskanjem v širino pri faktorju vejanja *b*?

_____ **b / (b – 1)** _____

(e) Kateri neinformirani algoritmi bi spremenili svojo rešitev, če bi poleg vozlišča **h** bilo ciljno vozlišče tudi **g**?

_____**DFS**_____

(f) Katero rešitev vrne algoritem A* in kakšna je njena cena?

_____ **a – c – e – f – h cena: 6**_____

(g) Kolikokrat algoritem IDA* ne uspe in mora povečati mejo F_b?

Uspe v PRVEM poskusu, meja F = 6

(h) Kakšna je f-ocena vozlišča **e**?

_____**f(e) = 5 + 4n (n>=0, ker lahko cikla)**_____

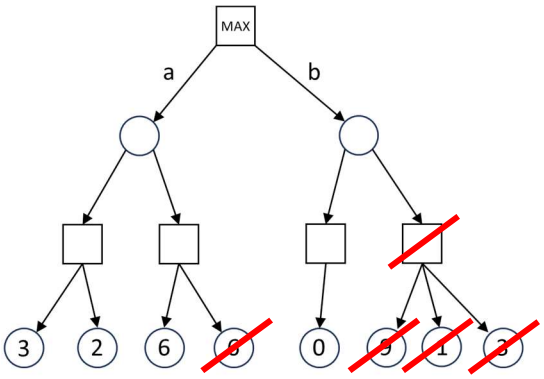
(i) Je podana hevristična funkcija dopustna? Če ni, predlagajte popravek/popravke!

_____**Ni dopustna. h(b) <= 4**_____

(j) Je podana hevristična funkcija monotona? Obrazložite!

____**Ni monotona, npr. f(c)=6 > f(e)=5, po poti a-c-e**_____

(2)
Podano je drevo igre v neki poziciji. Številke predstavljajo hevristične ocene pripadajočih vozlišč (pozicij).



(a) Kakšni sta oceni potez a in b in katero potezo izbere algoritem minimax?

a: **3** b: **0** izbrana: **a**

(b) Koliko vozlišč bi prihranilo (porezalo) alfa-beta rezanje v primerjavi z navadnim algoritmom minimax? Štejte vozlišča na vseh nivojih skupaj.

5 vozlišč (glej prečrtane na sliki)

(3)
Imamo naslednjo igro za dva igralca s popolno informacijo. Z mize jemljemo žetone. Igralec na potezi lahko vzame en ali dva žetona. Zmaga tisti igralec, ki mizo pusti prazno, torej pobere zadnji žeton (ali zadnja dva). Trenutno igro začnemo s petimi žetoni na mizi. Namig: pri spodnjih vprašanjih si lahko pomagate tako, da narišete celotno drevo igre, v končnih stanjih pa rezultat določajo pravila.

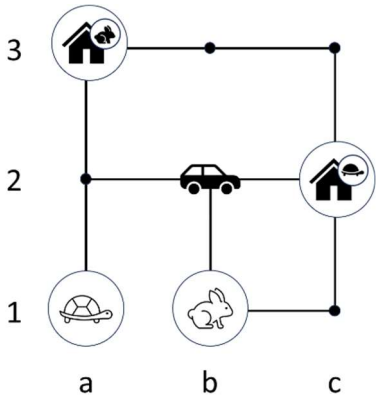
(a) Katero potezo (vzemi enega ali vzemi dva) bi izbral algoritem minimax? Ali sta obe potezi enakovredni?

Nista enakovredni, izbere »vzemi 2«, ki zmaga

(b) Zdaj pa si pomagamo z Monte Carlo algoritmom. Iz stanja, ko je prvi igralec vzel enega, so bile odigrane naslednje simulacije {[2,2], [1,2,1], [2,2], [1,1,2], [1,1,1,1]}, iz stanja, ko je prvi igralec vzel dva žetona, pa te {[1,2], [1,1,1], [1,1,1], [2,1], [1,1,1]}. Oznaka 1 oz. 2 pomeni koliko žetonov je vzel igralec, ki je bil na potezi. Simulacija je vedno potekala do prazne mize (konca igre), kjer vemo točen rezultat. Kolikšni sta vrednosti zmag/porazov za obe stanji in katero potezo bi izbral algoritem Monte Carlo?

1: **prvi 3 zmage, drugi 2 zmagi** 2: **prvi 2 zmagi, drugi 3 zmage** izbrana poteza: **1 (izgubi)**

(4)
Slika na desni predstavlja situacijo v kateri sta se dva hišna ljubljenčka, želva in zajec, nesrečno izgubila. Z avtomobilom ju želimo poiskati in vrniti domov; želva stanuje na polju **c2**, zajec pa na polju **a3**. V avtomobilu je zadosti prostora za oba, če bi ju želeli peljati skupaj, stepla se pa tudi ne bosta. Povezave predstavljajo cesto; direktne poti med **a1** in **b1** ni, prav tako je ni med **b2** in **b3**. Svet opišemo z relacijo *at(object, location)*, kjer je *object* lahko avto, želva ali zajec, *location* pa je lahko polje na zemljevidu (npr. **b2**) ali za ljubljenčka tudi »*in car*«. Akcije v tem svetu so *left/right/up/down*, kar ustrezno spremeni pozicijo avtomobila, *pick(X)*, ki pobere ljubljenčka *X* in *drop(X)*, ki ljubljenčka *X* odloži. Seveda se mora avtomobil nahajati na ustrezni lokaciji, da lahko nekoga pobere, odloži pa ga načeloma lahko kjerkoli. Še to: optimalen voznik bo potreboval manj kot 10 premikov avtomobila.



(a) Dopolnite spodnji STRIPS opis akcije *pick(X)*, torej akcije, ki pobere ljubljenčka *X*.

preconditions: at(car, Loc), **at(X, Loc)** effects: ~at(X, Loc), **at(X, in_car)**

(b) Dopolnite spodnji STRIPS opis akcije *drop(X)*, torej akcije, ki odloži ljubljenčka *X*.

preconditions: at(car, Loc), **at(X, in_car)** effects: ~at(X, 'in car'), **at(X, Loc)**

(c) Katero akcijo oz. akcije bo planer po principu regresiranja ciljev upošteval kot prvo?

drop(turtle, c2), drop(rabbit, a3)

(d) Kateri plan bo vrnil planer po principu regresiranja ciljev, če uporablja iterativno poglabljanje? (lahko je več rešitev)

D Pick U L D Pick U U Drop(rabbit) R R D Drop(turtle)

(e) Trenutna množica ciljev je G. Napišite formulo za regresiranje teh ciljev skozi akcijo A? **G U can(A) – adds(A)**