

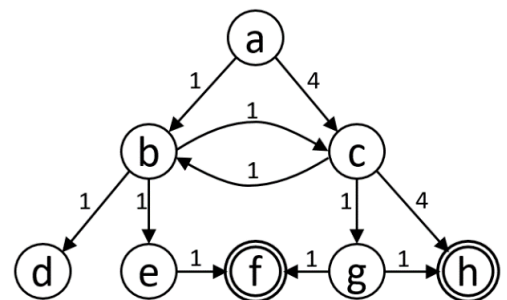
Literatura (prosojnice, knjige, zapiski, elektronski pripomočki) ni dovoljena.
Vsaka naloga je vredna 10 točk. Vsako nalogo rešujte v predvidenem prostoru.
Če rešitev rešite na pomožni list, jasno označite, na katero nalogo se nanaša.
Iz vaše rešitve mora biti viden postopek reševanja.
Podpišite se na vse liste, ki jih oddate.
Na vprašanja odgovarjajte kratko (največ 2 povedi), daljši odgovori štejejo 0 točk.
Čas pisanja je 80 minut.

izpolni ocenjevalec

1	
2	
3	
4	
Σ	

1. NALOGA (10t):

Naj bo a začetno vozlišče preiskovanja, f in h pa sta ciljni vozlišči. Algoritmi preiskovanja naj generirajo naslednike vozlišč po abecednem vrstnem redu. Npr. vrstni red naslednikov vozlišča c je: b, g, h. Vsi preiskovalni algoritmi naj tudi razpoznavajo cikle in generirano vozlišče, ki sklene cikel, takoj zavržejo. Vendar pa obravnavajo graf kot drevo. Torej, če pridejo do kakega vozlišča N po različnih poteh, naredijo kopijo N' vozlišča N in obravnavajo N', kot da bi bilo novo vozlišče.



- a) (4t) Katero rešitveno pot najde algoritem A*, če so hevrstične ocene vozlišč pa podane v tabeli? Pri rešitvi zapiši tudi vrstni red generiranih vozlišč.

X	a	b	c	d	e	f	g	h
h(X)	4	6	2	6	3	0	7	0

- b) (2t) Ali je hevrstika h iz prejšnje točke dopustna? Če ni, kako je potrebno popraviti vozlišča?

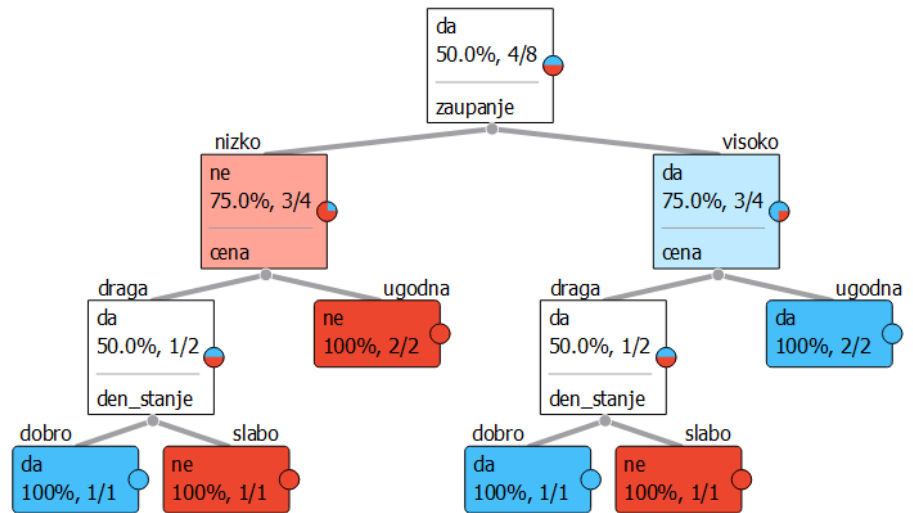
- c) (2t) V nalogi je potrebno popraviti hevrstično oceno samo enega vozlišča, da algoritem A* hitreje najde optimalno pot. Katero vozlišče je to in kako je potrebno spremeniti njegovo hevrstično oceno?

- d) (2t) V kateri podatkovni strukturi je smiselno hraniti podatke o vozliščih (ime vozlišča in vrednost cenitvene funkcije f), da je lahko implementacija preiskovanja narejena učinkovito?

2. NALOGA (10t):

Podano je odločitveno drevo na sliki, ki ga uporabljamo za odločanje o nakupu valute Bitcoin. Drevo je zgrajeno iz učnih podatkov, ki imajo attribute: *zaupanje* (zaupanje v prodajalca – nizko ali visoko), *cena* (nakupna cena – ugodna ali draga) in *den_stanje* (lastno denarno stanje – slabo ali dobro). Razred je spremenljivka *nakup*, ki ima lahko vrednosti "da" (kupimo) ali "ne" (ne kupimo).

Legenda: Vozlišča v drevesu prikazujejo razred (da/ne), delež večinskega razreda in število primerov, ki pripadajo razredu da/ne.



Naloge:

- a) (2t) Iz drevesa na skici rekonstruiraj učno množico in jo zapiši v tabelo na desni.

zaupanje	cena	den_stanje	nakup

- b) (3t) Izračunaj informacijski prispevek vseh možnih atributov primerov, ki jih predstavlja levo poddrevo korena. Na drugem nivoju drevesa so ti primeri nadalje razdeljeni glede na vrednost atributa *cena*. Ali je to optimalno glede na informacijski prispevek? Utemelji.

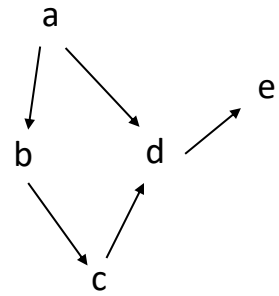
- c) (3t) Z uporabo rezalne množice na desni strani poreži zgornje drevo s postopkom zmanjševanja napake (REP). Rezanje prikaži na zgornji skici drevesa.

zaupanje	cena	den_stanje	nakup
nizko	draga	dobro	ne
nizko	ugodna	dobro	da
visoko	draga	slabo	da

- d) (2t) Zakaj je pri postopku REP iz prejšnje naloge potrebna posebna rezalna množica?

3. NALOGA (10t):

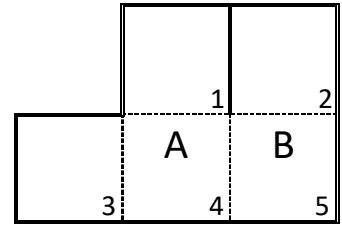
Podana je bayesovska mreža, ki je prikazana na sliki. Predpostavimo, da vse povezave v mreži predstavljajo pozitivno vzročnost med dogodki. Odgovori na naslednja vprašanja:



- a) (1t) Zapiši vse množice vozlišč, ki d-ločujejo a in c.
- b) (1t) Predpostavi, da mrežo dopolnimo tako, da dodamo povezavo iz vozlišča e v vozlišče a (dopolnitev velja samo za to podnalogo, ostale reši na originalni mreži). Ponovi nalogo iz prejšnje točke: zapiši vse množice vozlišč, ki d-ločujejo a in c.
- c) (2t) Zapiši vse načine, kako lahko obrnemo vstopne/izstopne povezave vozlišča b, da dobimo mrežo, ki je l-ekvivalentna prvotni mreži.
- d) (1t) Če je možno, čim bolj poenostavi pogojni del v izrazu: $P(d|ab)$. Odgovor utemelji.
- e) (1t) Če je možno, čim bolj poenostavi pogojni del v izrazu: $P(d|abc)$. Odgovor utemelji.
- f) (4t) Izrazi verjetnost $P(a|c)$ z verjetnostmi, ki so podane v mreži.

4. NALOGA (10t):

Podan je majhen labirint, sestavljen iz petih sob (1, 2, 3, 4 in 5). V labirintu se nahajata dva robota (A in B). Stanja so opisana s predikatom $c(Room)$, kar pomeni, da je soba *Room* prazna, in s predikatom $in(Robot, Room)$, ki pove, da se robot *Robot* nahaja v sobi *Room*. Robota se lahko gibljeta le v smeri črtkanih povezav med sobami (dvojne obrobe predstavljajo stene).



Začetno stanje, ki je podano na sliki, je torej: $[c(1), c(2), c(3), in(A,4), in(B,5)]$

Podana je akcija $move(Robot, From, To)$, ki je definirana na naslednji način:

```
predpogoj: [in(Robot,From), c(To)]
omejitev: [robot(Robot), room(From), room(To), adjacent(From,To)]
add: [in(Robot,To), c(From)]
del: [in(Robot,From), c(To)]
```

- a) (2t) Definiraj akcijo $climbover(Robot1, Robot2, From, To)$, ki povzroči, da Robot1 spleza preko Robota2 in se s tem premakne s svojega polja na polje za robotom. Premik lahko izvede samo v ravni črti (ne okoli vogala), za kar po potrebi definiraj novi predikat.

- b) (2t) Zapiši vse možne primere uporabe akcije $climbover$.

- c) (3t) Kaj je rezultat regresiranja ciljev $[in(A,5), in(B,1)]$ skozi akcijo $move(A,4,5)$?

- d) (3t) Kaj je rezultat regresiranja ciljev $[in(A,5), in(B,4)]$ skozi akcijo $move(A,4,5)$?