

## FONDAMENTI DI INTELLIGENZA ARTIFICIALE

10 giugno 2021 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

**NOTA:** Consegnare la soluzione tramite un singolo file, che lo studente avrà cura di nominare come:

CognomeNomeDataAI

Ad esempio: RossiMario20210610AI

### Esercizio 1 (6 punti)

Si formalizzino in logica dei predicati del I ordine le seguenti frasi:

1. Se un qualunque alunno di una classe ha il COVID-19, la classe entra in quarantena.
2. Chiunque è positivo al test COVID-19 molecolare o (**OR NON ESCLUSIVO**) è positivo al test COVID-19 antigenico allora ha il COVID-19.
3. Giovanni è un alunno della classe primaA
4. Marco è un alunno della classe primaB
5. Marco è risultato positivo al test COVID-19 molecolare.

Si utilizzi la risoluzione per dimostrare che esiste almeno un alunno di una classe che entra in quarantena.

Si usino i seguenti predicati:

**alunnoClasse(Persona, Classe)**

**haCovid(Persona) .**

**quarantena(Classe) .**

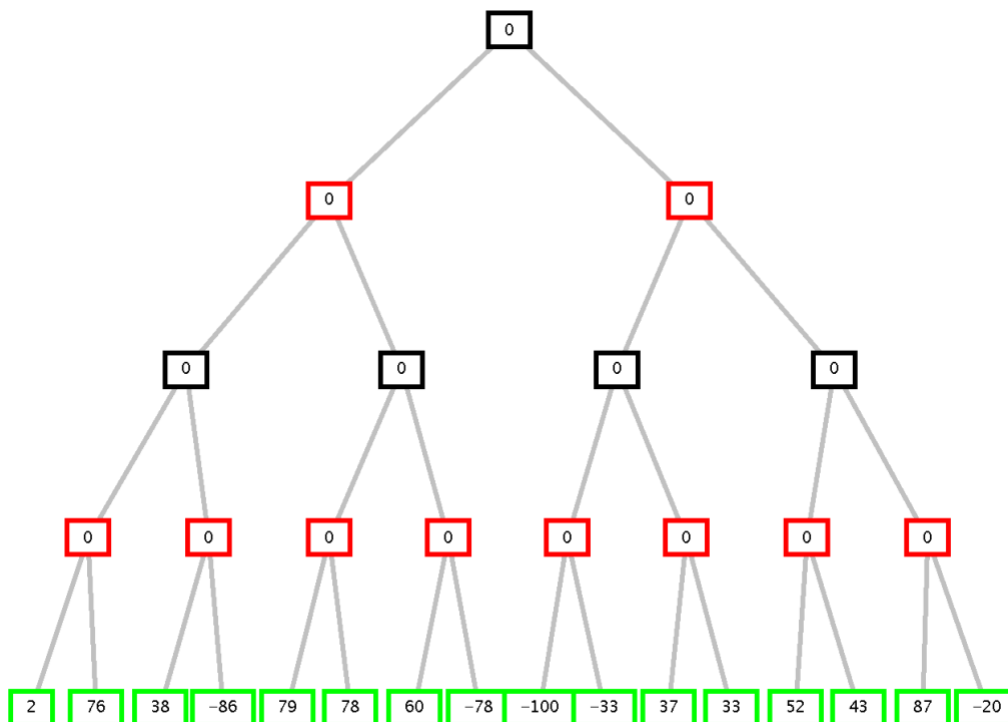
**testPosMol(Persona) .**

**testPosAntig(Persona) .**

### Esercizio 2 (4 punti)

Si consideri il seguente albero di gioco in cui il primo giocatore è MAX.

- a) Si indichi come l'algoritmo min-max risolve il problema indicando il valore con cui viene etichettato il nodo iniziale e la mossa selezionata dal primo giocatore (nome dell'arco dal nodo radice).
- b) Si mostrino poi i tagli che l'algoritmo alfa-beta consente indicando gli archi che verranno tagliati. Si indichino i nomi degli archi iniziando con la lettera "a" e facendola seguire con un numero crescente da sinistra a destra e dall'alto al basso. Ad esempio, i due archi che si dipartono dalla radice saranno nominati a1 (quello più a sinistra) e a2. L'arco che connette il nodo foglia più a sinistra (con valore 2) sarà denominato a15, mentre l'ultimo arco che connette il nodo foglia più a destra (valore -20) a30.



### Esercizio 3 (6 punti)

Si consideri il seguente CSP che lega le variabili A, B, C, D:

A::[4, 5, 6, 7, 8, 9, 10]

$A \geq B - 7$

B::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

$C \geq B - 5$

C::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

$A = D + 5$

D::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Durante la ricerca, fino alla prima soluzione, si applichi il Forward Checking dopo ogni passo di labeling. Nella scelta della prossima variabile da istanziare applicare l'euristica **Minimum Remaining Value** (poi, a parità di cardinalità di dominio, scegliere in base all'ordine alfabetico dei nomi delle variabili) e per il labeling si considerino i valori di dominio in ordine crescente, partendo dal più piccolo. Si mostri come si raggiunge la soluzione indicando ad ogni passo i valori della variabili via via istanziate (labeling), i domini delle variabili non ancora istanziate (eventualmente ridotti a causa del forward checking) e l'eventuale presenza di backtracking.

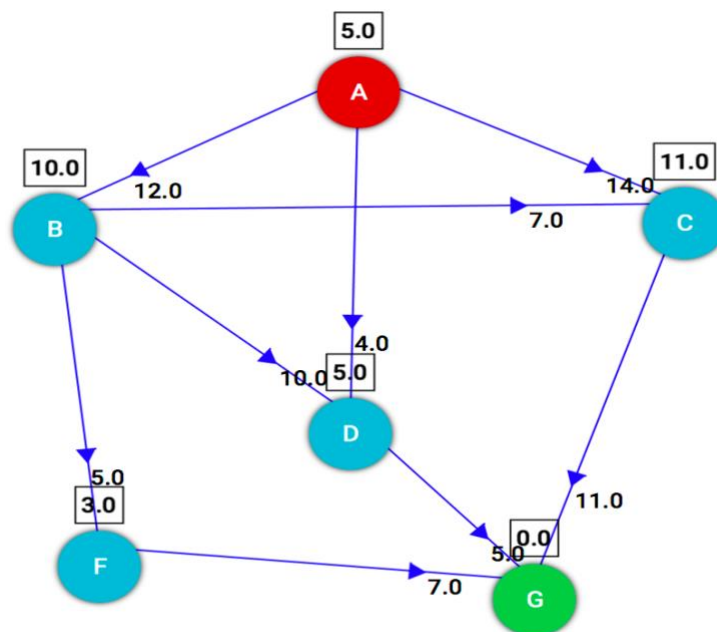
#### Esercizio 4 (5 punti)

Si realizzi un predicato Prolog `newList(Start, End, L1)` che crei una lista di interi `L1` contenente i soli valori della lista `Start` che sono strettamente maggiori di tutti i valori della lista `End`. Se non ne esistono o `Start` è vuota dovrà restituire la lista vuota. A tal fine si definisca e si utilizzi anche un predicato `checkMax(List, N)` che data una lista `List` e un numero intero `N`, abbia successo se `N` è strettamente maggiore di tutti gli elementi della lista `List`. Di seguito si riporta qualche esempio di esecuzione Prolog:

```
?- newList([7,4,8,1],[4,2,5,6],L1).
L1 = [7, 8]
?- newList([7,4,8,5,6],[ ],L1).
L1 = [7, 4, 8, 5, 6]
?- newList([7,4,8,5,6],[10],L1).
L1 = [ ]
```

#### Esercizio 5 (7 punti)

Si consideri il seguente grafo, dove A è il nodo iniziale e G il nodo goal, e il numero associato agli archi è il costo dell'operatore per andare dal nodo di partenza al nodo di arrivo dell'arco. Vicino ad ogni nodo, in un quadrato, è indicata inoltre la stima euristica della sua distanza dal nodo goal G:



- Si applichi la ricerca **Depth-first** su alberi (che non tiene traccia dei nodi già visitati) e si indichino i nodi espansi nell'ordine di espansione. In caso di non-determinismo (più figli di un nodo), si scelga il nodo da espandere in base all'ordine alfabetico del nome. Non si consideri l'euristica  $h(n)$  indicata nel quadrato a fianco di ogni nodo in figura e neppure il costo degli archi, ma solo la profondità.
- Si applichi poi la ricerca **A\*** su alberi (che non tiene traccia dei nodi già visitati) e si indichino i nodi espansi nell'ordine di espansione. In caso di non-determinismo si scelga il nodo da espandere in base all'ordine alfabetico.
- $h(n)$  è ammissibile? Si motivi la risposta.
- Si considerino i costi degli archi e si commentino i costi delle soluzioni delle due strategie (depth-first e A\*) in termini di ottimalità.

#### Esercizio 6 (4 punti)

Si descriva sinteticamente l'unificazione, dove è utilizzata in Prolog, cos'è l'occur-check e cosa implichi il suo non utilizzo in Prolog.

## 10 giugno 2021 - Soluzioni

### Esercizio 1

1.  $\forall \text{Classe}, \forall \text{Persona} \text{ alunnoClasse}(\text{Persona}, \text{Classe}) \wedge \text{haCovid}(\text{Persona}) \rightarrow \text{quarantena}(\text{Classe}).$
  2.  $\forall \text{Persona} (\text{testPosMol}(\text{Persona}) \vee \text{testPosAntig}(\text{Persona}) ) \rightarrow \text{haCovid}(\text{Persona}).$
  3.  $\text{alunnoClasse}(\text{giovanni}, \text{primaA}).$
  4.  $\text{alunnoClasse}(\text{marco}, \text{primaB}).$
  5.  $\text{testPosMol}(\text{marco}).$
- Query:  $\exists X \exists Y \text{ alunnoClasse}(X, Y) \wedge \text{quarantena}(Y).$

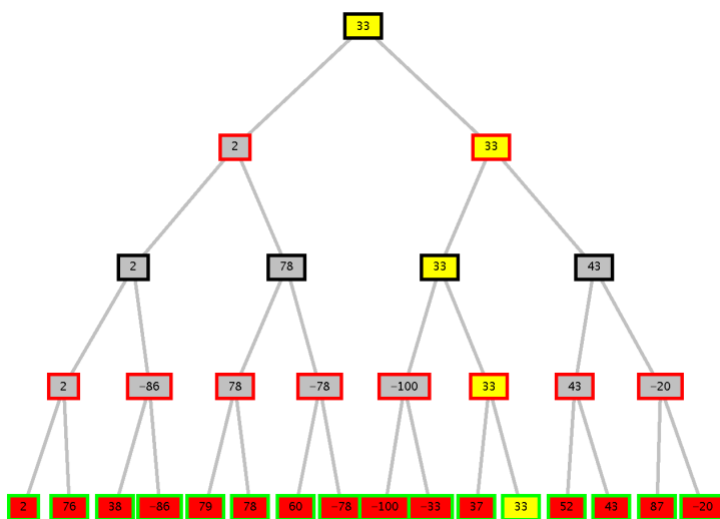
*Trasformazione in clause:*

- C1:  $\neg \text{alunnoClasse}(\text{Persona}, \text{Classe}) \vee \neg \text{haCovid}(\text{Persona}) \vee \text{quarantena}(\text{Classe}).$   
 C2a:  $\neg \text{testPosMol}(\text{Persona}) \vee \text{haCovid}(\text{Persona}).$   
 C2b:  $\neg \text{testPosAntig}(\text{Persona}) \vee \text{haCovid}(\text{Persona}).$   
 C3:  $\text{alunnoClasse}(\text{giovanni}, \text{primaA})$   
 C4:  $\text{alunnoClasse}(\text{marco}, \text{primaB})$   
 C5:  $\text{testPosMol}(\text{marco})$   
 QNeg:  $\neg \text{quarantena}(Y) \vee \neg \text{alunnoClasse}(X, Y).$

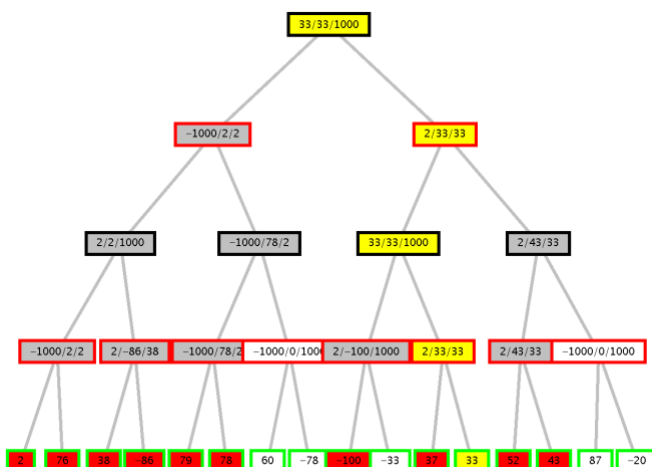
*Risoluzione:*

- C6: QNeg+C1:  $\neg \text{alunnoClasse}(\text{Persona}, \text{Classe}) \vee \neg \text{haCovid}(\text{Alunno})$   
 C7: C6+C4:  $\neg \text{haCovid}(\text{marco})$   
 C8: C7+C2a:  $\neg \text{testPosMol}(\text{marco})$   
 C9: C8 + C5      **contraddizione!!**

**Esercizio 2.** Min-max: strada in giallo (arco a2) – valore nodo radice 33.



Alfa-beta: In rosso i nodi espansi, in giallo la strada trovata, i nodi in bianco non sono esplorati per effetto dei tagli.



Archi tagliati a10, a14, a24.

Scelta per il ramo a destra (a2), valore propagato 33.

### Esercizio 3

Con euristica MRV:

A::[4, 5, 6, 7, 8, 9, 10]

B::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

C::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

D::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

A>=B-7

C>=B-5

A=D+5

	A	B	C	D
Labeling	<b>A=4</b>	[1...10]	[1...10]	[1...10]
FC - Backtracking	A=4	[1...10]	[1...10]	<b>Fail</b>
Labeling	<b>A=5</b>	[1...10]	[1...10]	[1...10]
FC - Backtracking	A=5	[1...10]	[1...10]	<b>Fail</b>
Labeling	<b>A=6</b>	[1...10]	[1...10]	[1...10]
FC	A=6	[1...10]	[1...10]	<b>[1]</b>
Labeling	A=6	[1...10]	[1...10]	<b>D=1</b>
FC	A=6	[1...10]	[1...10]	D=1
Labeling	A=6	<b>B=1</b>	[1...10]	D=1
FC	A=6	B=1	[1...10]	D=1
Labeling	A=6	B=1	<b>C=1</b>	D=1

### Esercizio 4

```
newList([],_,[]):-!.
```

```
newList(Start,[],Start):-!.
```

```
newList([Hs|Ss],End,[Hs|S1]):-
```

```
    checkMax(End,Hs),!,newList(Ss,End,S1).
```

```
newList([_|Ss],End,S1):-
```

```
    newList(Ss,End,S1).
```

```
checkMax([],_).
```

```
checkMax([Hs|Ss],N):- N > Hs, checkMax(Ss,N).
```

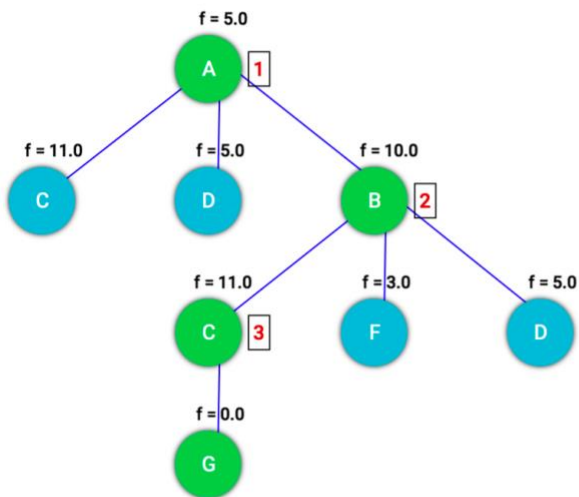
### Esercizio 5

Con ricerca Depth-first e nodi espansi ABC si trova la soluzione (non ottimale):

ABCG con costo 30.

Con ricerca A\* e nodi espansi AD si trova la soluzione trovata ADG è a costo inferiore 9 (ottimale perché l'euristica è ammissibile).

Per le altre domande si consulti il materiale del corso.

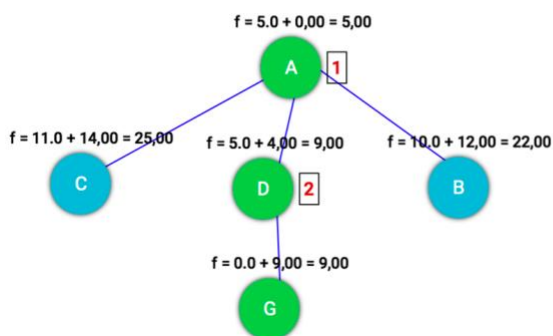


#### Operations

Show operations

- 1) A [f = 5.0]
- 2) B [f = 10.0]
- 3) C [f = 11.0]
- /) C [f = 11.0]
- /) D [f = 5.0]
- /) G [f = 0.0]
- /) F [f = 3.0]
- /) D [f = 5.0]

Path cost: 30.0  
Nodes expanded: 3  
Queue size: 4  
Max queue size: 5



#### Operations

Show operations

- 1) A [f = 5.0 + 0.00 = 5.00]
- 2) D [f = 5.0 + 4.00 = 9.00]
- /) C [f = 11.0 + 14.00 = 25.00]
- /) G [f = 0.0 + 9.00 = 9.00]
- /) B [f = 10.0 + 12.00 = 22.00]

Path cost: 9.0  
Nodes expanded: 2  
Queue size: 2  
Max queue size: 3

### Esercizio 6

Si veda il materiale del corso.