

**NOTA:** Consegnare la soluzione tramite un singolo file, che lo studente avrà cura di nominare come: CognomeNomeDataAI. Ad esempio: RossiMario20210709AI

### Esercizio 1 (6 punti)

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

1. La professoressa (prof) premia tutti gli studenti appartenenti alle squadre.
2. Ogni studente appartiene a una squadra.
3. Tutti gli studenti premiati dalla professoressa sono bravi.
4. Mario è uno studente.

Si dimostri poi, tramite il principio di risoluzione, che esiste almeno uno studente bravo. Si usino i seguenti predicati con l'ovvio significato:

**premia (X, Y)** – X premia Y;

**studente (X)** – X è uno studente;

**squadra (X, Y)** – X appartiene alla squadra Y;

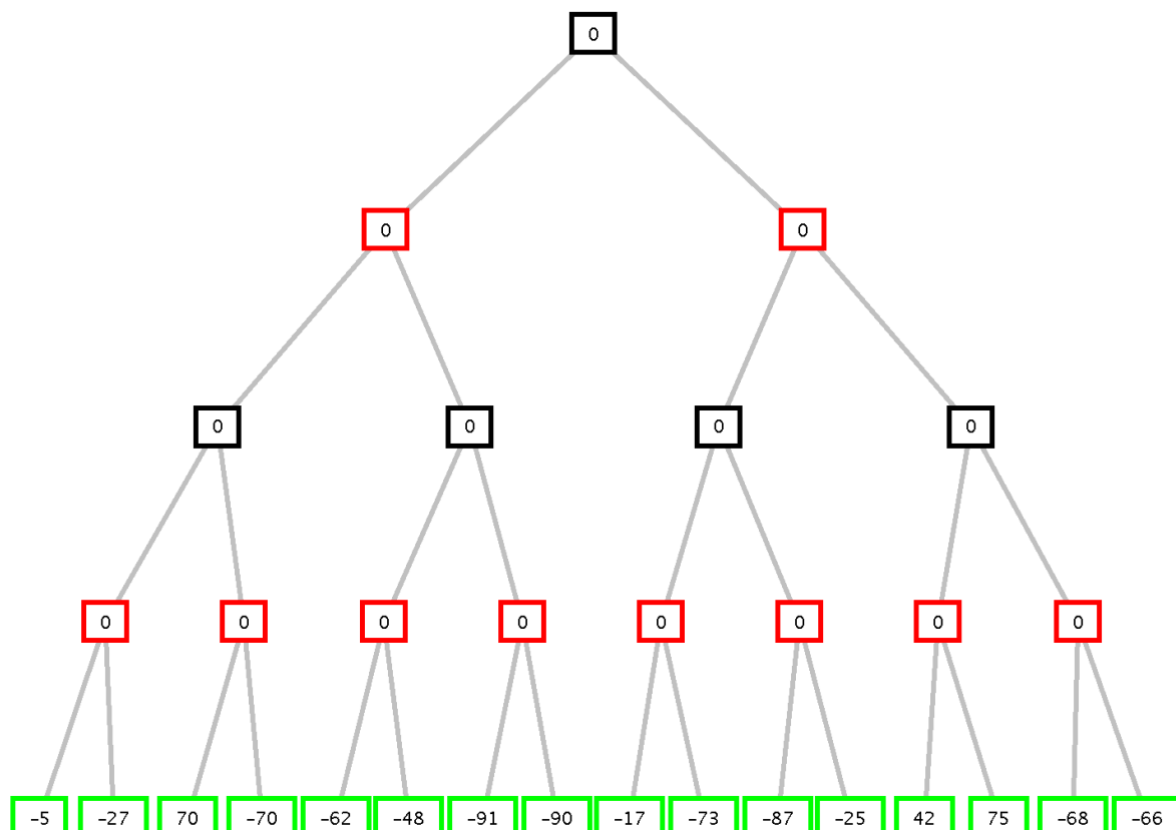
**bravo (X)** X è bravo.

I termini "prof", "mario" siano considerati costanti.

### 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 (arco a1, o a2).
- 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 -5) sarà denominato a15, mentre l'ultimo arco che connette il nodo foglia più a destra (valore -66) a30.



### Esercizio 3 (7 punti)

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

A::[1, 2, 3, 4, 5, 6]

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

C::[1, 2, 3, 4, 5, 6]

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

$A \geq B - 8$

$C = B - 5$

$A \leq D - 5$

Si applichi, durante la ricerca fino alla prima soluzione, il Forward Checking dopo ogni passo di labeling, con le due euristiche su scelta della prossima variabile da istanziare:

- Si applichi labeling e FC, considerando le variabili secondo il loro **ordinamento alfabetico**;
- Si applichi labeling e FC, considerando, nella scelta della prossima variabile da istanziare, l'euristica **Minimum Remaining Value** (poi, a parità di cardinalità di dominio, scegliere in base all'ordine alfabetico dei nomi delle variabili). Quanto cambia la ricerca e i passi svolti in questo caso rispetto al precedente a)?

Nel labeling, per il valore da assegnare alla variabile, si considerino in entrambi i casi i valori di dominio in ordine crescente, partendo dal più piccolo.

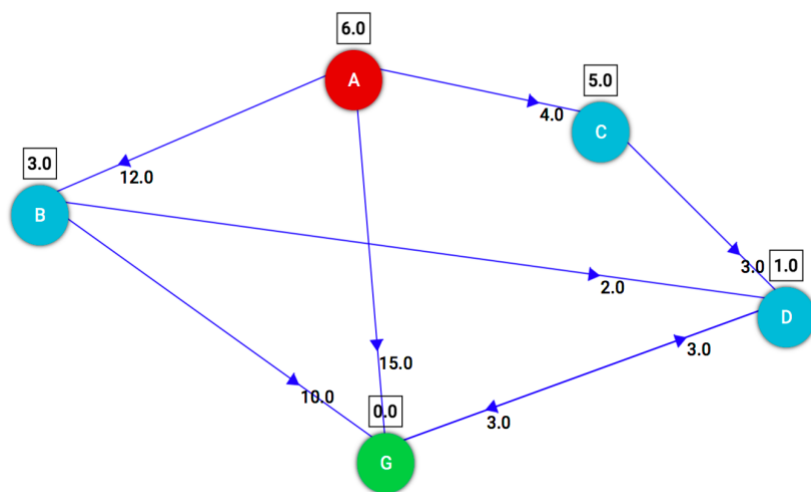
### Esercizio 4 (5 punti)

Si realizzi un predicato Prolog `multiList(L, L1, N)` che restituisca una nuova lista L1 nelle quale gli elementi sono liste. Ciascuna di esse dovrà contenere un elemento di L ripetuto N volte. A tal fine si definisca e si utilizzi anche un predicato `buildList(X, Y, N)` che dato un elemento X e un numero naturale N, dia in uscita una lista Y in cui l'elemento X è ripetuto N volte. Di seguito si riporta qualche esempio di esecuzione Prolog:

```
?- multiList([a, b, c, d], Y, 0).  
Y = [[], [], [], []]  
?- multiList([a, b, c, d], Y, 3).  
Y = [[a, a, a], [b, b, b], [c, c, c], [d, d, d]]  
?- multiList([a, b, c, d], Y, 1).  
Y = [[a], [b], [c], [d]]  
?- multiList([], Y, 3).  
Y = []  
?- buildList(a, Y, 3).  
Y = [a, a, a]
```

### Esercizio 5 (6 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 **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 del nome.
- $h(n)$  è ammissibile? Si motivi la risposta.
- Si indichi il costo della soluzione, motivando se è garantita o meno l'ottimalità.

### Esercizio 6 (4 punti)

Si descrivano le tecniche di node-, arc- e n-consistenza su un grafo CSP.

## 9 luglio 2021 - Soluzioni

### Esercizio 1

#### Trasformazione in clausole

1.  $\forall X \forall Y \text{ studente}(X) \wedge \text{squadra}(X, Y) \rightarrow \text{premia}(\text{prof}, X).$   
 $\neg \text{studente}(X) \vee \neg \text{squadra}(X, Y) \vee \text{premia}(\text{prof}, X)$
2.  $\forall X \text{ studente}(X) \rightarrow \exists S \text{ squadra}(X, S).$   
 $\forall X \exists S \neg \text{studente}(X) \vee \text{squadra}(X, f(X))$  Skolem
3.  $\forall X \forall Y, \text{ studente}(X) \wedge \text{premia}(\text{prof}, X) \rightarrow \text{bravo}(X).$   
 $\neg \text{studente}(X) \vee \neg \text{premia}(\text{prof}, X) \vee \text{bravo}(X)$
4.  $\text{studente}(\text{mario})$

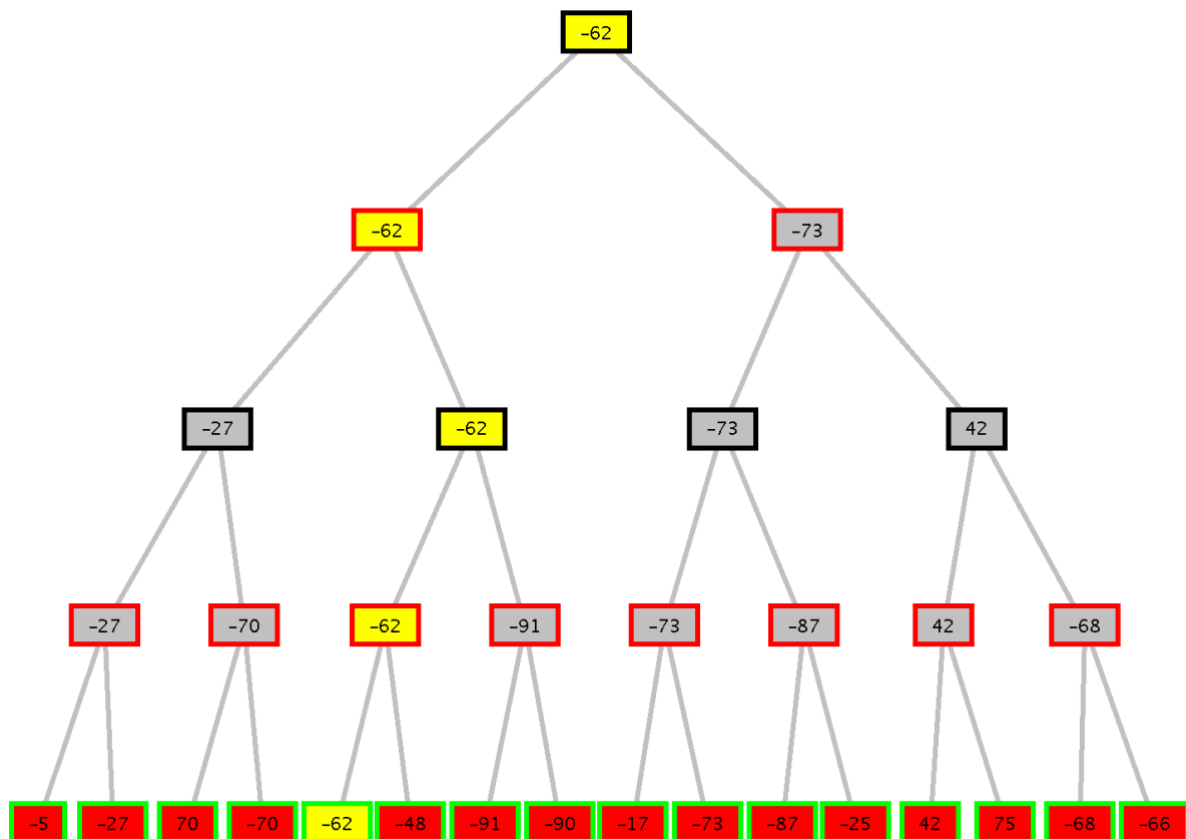
G.  $\exists X \text{ studente}(X) \wedge \text{bravo}(X)$   
 Gneg.  $\neg \text{studente}(X) \vee \neg \text{bravo}(X)$

#### Risoluzione

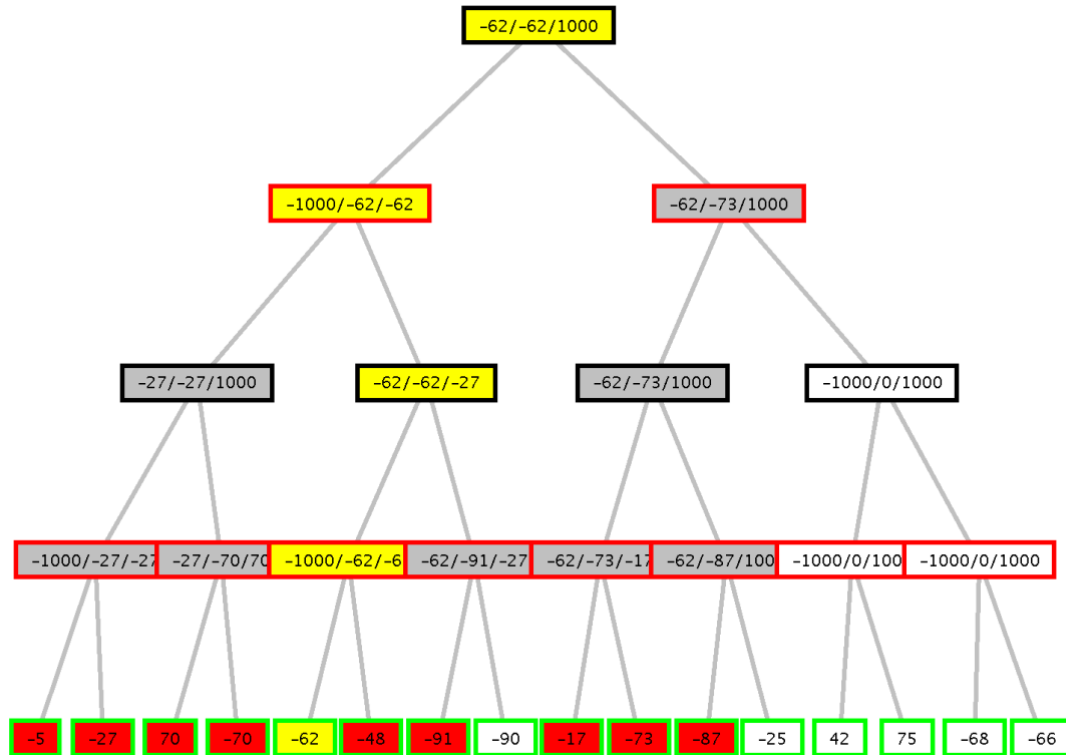
5. **Gneg + 3:**  $\neg \text{studente}(X) \text{ or } \neg \text{premia}(\text{prof}, X)$
6. **5 + 1:**  $\neg \text{studente}(X) \text{ or } \neg \text{squadra}(X, Y)$
7. **6 + 2:**  $\neg \text{studente}(X)$
8. **7 + 4:** *clausola vuota*

### Esercizio 2

Min-max: strada in giallo – valore nodo radice -62, ramo a1.



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



Archi tagliati a6, a22, a26.

Scelta per il ramo a1, valore propagato -62.

### Esercizio 3

Con scelta variabili in base al nome:

	A	B	C	D
Labeling	<b>A=1</b>	[1...10]	[1...6]	[1...10]
FC	A=1	[1...9]	[1...6]	[6...10]
Labeling	A=1	<b>B=1</b>	[1...6]	[6...10]
FC e Backtracking	A=1	B=1	<b>Fail</b>	[6...10]
Labeling	A=1	<b>B=2</b>	[1...6]	[6...10]
FC e Backtracking	A=1	B=2	<b>Fail</b>	[6...10]
Labeling	A=1	<b>B=3</b>	[1...6]	[6...10]
FC e Backtracking	A=1	B=3	<b>Fail</b>	[6...10]
Labeling	A=1	<b>B=4</b>	[1...6]	[6...10]
FC e Backtracking	A=1	B=4	<b>Fail</b>	[6...10]
Labeling	A=1	<b>B=5</b>	[1...6]	[6...10]
FC e Backtracking	A=1	B=5	<b>Fail</b>	[6...10]
Labeling	A=1	<b>B=6</b>	[1...6]	[6...10]
FC	A=1	B=6	[1]	[6...10]
Labeling e FC	A=1	B=6	<b>C=1</b>	[6...10]
Labeling	A=1	B=6	C=1	<b>D=6</b>
<b>Soluzione</b>	<b>A=1</b>	<b>B=6</b>	<b>C=1</b>	<b>D=6</b>

Con euristica MRV:

	A	B	C	D
Labeling	<b>A=1</b>	[1...10]	[1...6]	[1...10]
FC	A=1	[1...9]	[1...6]	[6...10]
Labeling	A=1	[1...9]	[1...6]	<b>D=6</b>
FC	A=1	[1...9]	[1...6]	D=6
Labeling	A=1	[1...9]	<b>C=1</b>	D=6
FC	A=1	[6]	C=1	D=6
Labeling	A=1	<b>B=6</b>	C=1	D=6
<b>Soluzione</b>	<b>A=1</b>	<b>B=6</b>	<b>C=1</b>	<b>D=6</b>

Con euristica MRV si arriva direttamente alla prima soluzione, senza backtracking.

#### Esercizio 4

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

```
multiList([X|Xs], [Y|Ys], N) :- buildList(X, Y, N), multiList(Xs, Ys, N).
```

```
buildList(_, [], 0) :- !.
```

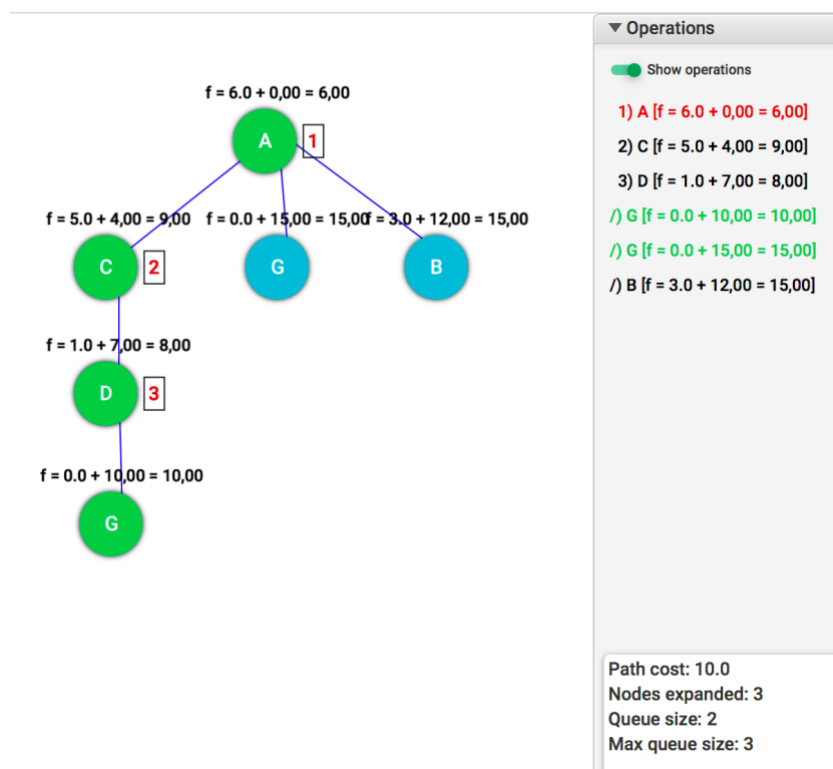
```
buildList(X, [X|Ys], N) :- N1 is N - 1, buildList(X, Ys, N1).
```

#### Esercizio 5

Con nodi espansi ACD si trova la soluzione (ottimale):

ACDG con costo 10.

Con A\*, la soluzione trovata è ottimale perché l'euristica è ammissibile.



#### Esercizio 6

Si vedano le slide del corso.