FONDAMENTI DI INTELLIGENZA ARTIFICIALE

23 Gennaio 2020 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (6 punti)

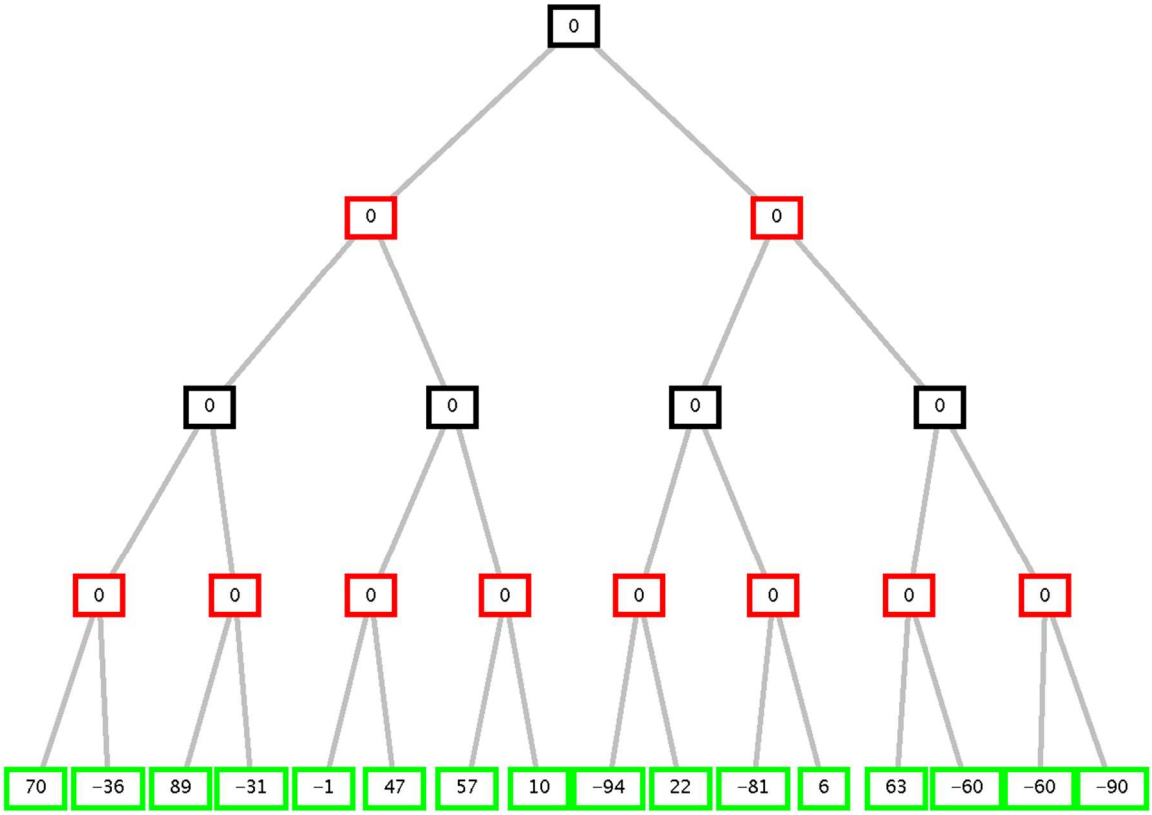
Si formalizzino le seguenti frasi in logica dei predicati:

1. I cani sono animali.
2. Qualunque cane che abbaia spaventa.
3. Qualunque animale che spaventa non è amato.
4. Esiste un cane che abbaia.

usando i seguenti predicati: **`cane(X)`** (X è un cane), **`animale(X)`** (X è un animale), **`abbaia(X)`** (X abbaia), **`spaventa(X)`** (X spaventa, è spaventoso), **`amato(X)`** (X è amato). Le si trasformi in clausole e si usi poi il principio di risoluzione per dimostrare che c'è un animale che non è amato.

Esercizio 2 (5 punti)

Si consideri il seguente albero di gioco in cui il primo giocatore è *MAX*. Si mostri come l'algoritmo *min-max* e l'algoritmo *alfa-beta* risolvono il problema e la mossa selezionata dal primo giocatore.



Esercizio 3 (6 punti)

Dato il seguente programma Prolog, `aggiungi(L,B,L1)` che, data la lista di liste `L`, e il numero intero `B`, produce in uscita la lista di liste `L1` in cui ad ogni lista elemento di `L` che non contenga già `B`, è stato aggiunto il numero `B` in testa:

```
aggiungi([],_,[]).
aggiungi([H|T],B,[H|T1]):- member(B,H), !,
                           aggiungi(T,B,T1).
aggiungi([H|T],B,[[B|H]|T1]):- aggiungi(T,B,T1).

member(X, [X|_]):-!.
member(X, [_|T]) :- member(X, T).
```

disegnare l'albero SLD per il goal seguente (si indichino i tagli effettuati dal *cut* e non si espandano gli eventuali rami tagliati):

```
?-aggiungi ([[1,5],[1]],5, L1).
```

Esercizio 4 (5 punti)

Si consideri una lista `L` formata, a sua volta, da liste di numeri interi. Scrivere un programma PROLOG `cresceSum(L)` che, data la lista di liste `L`, abbia successo se la somma degli elementi di ciascuna lista di `L` forma un ordinamento strettamente crescente.

Ad esempio alla query:

```
?- cresceSum([[1,5],[1,7]])
```

la risposta sarà:

```
Yes
```

in quanto la somma degli elementi della prima lista [1,5] è 6 e la somma degli elementi della seconda lista [1,7] è 8 e 6< 8.

Mentre alla query:

```
?- cresceSum([[1,5],[1,3]])
```

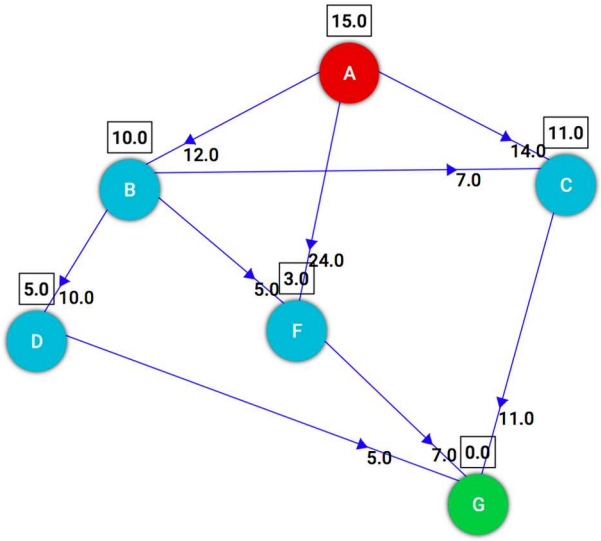
la risposta sarà:

```
No
```

in quanto la somma degli elementi della prima lista [1,5] è 6 mentre la somma degli elementi della seconda lista [1,3] è 4 e 6 > 4.

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:



a) Si applichi la ricerca A* e si disegni l'albero di ricerca sviluppato indicando per ogni nodo *n* l'ordine di espansione. In caso di non-determinismo, si scelgano i nodi da espandere in base all'ordine alfabetico del loro nome. Si consideri come euristica h(*n*) quella indicata nel quadrato a fianco di ogni nodo in figura.
b) L'euristica è ammissibile?
c) Qual è il costo di cammino trovato da A* ed il numero di nodi espansi per arrivare al goal G a partire dal nodo iniziale A?

Esercizio 6 (3 punti)

Si confrontino in termini di complessità spaziale, temporale e ottimalità gli algoritmi di ricerca non-informata *depth-first*, *breadth first* e ad *approfondimento iterativo*, immaginando che il costo degli archi sia sempre uguale a 1.

23 Gennaio 2020 - Soluzioni

Esercizio 1

1. $\forall X$ cane(X) $\longrightarrow$ animale(X)
2. $\forall X$ cane(X) $\wedge$ abbaia(X) $\longrightarrow$ spaventa(X)
3. $\forall X$ animale(X) $\wedge$ spaventa(X) $\longrightarrow$ $\neg$amato(X)
4. $\exists X$ cane(X) $\wedge$ abbaia(X)

   Goal: $\exists X$ animale(X) $\wedge$ $\neg$amato(X)

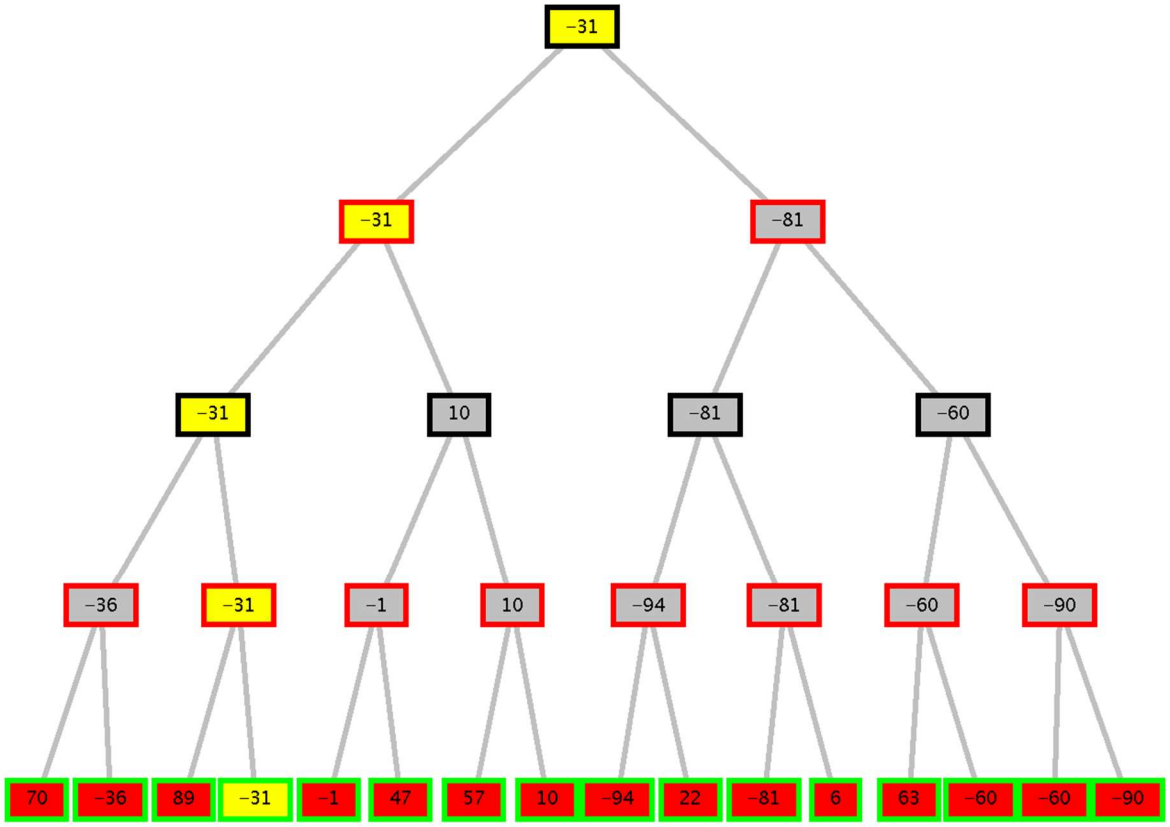
Clausole:

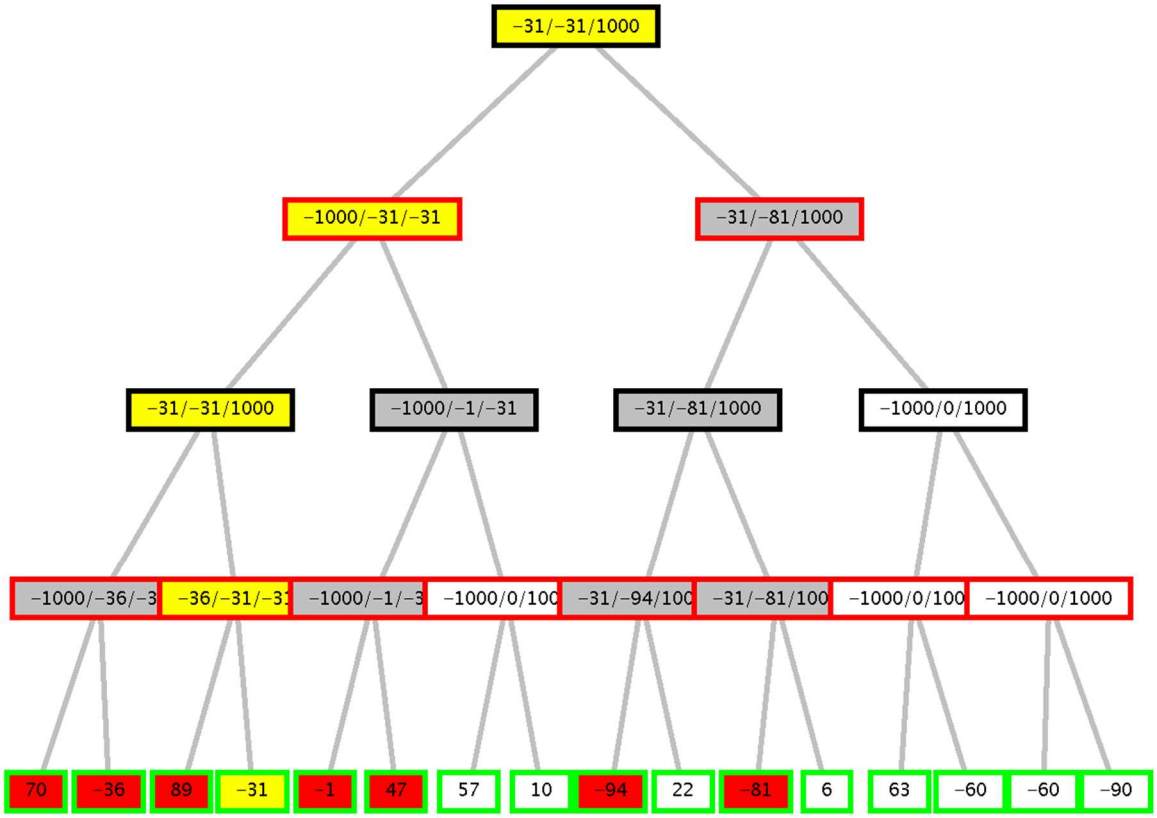
1. $\neg$ cane(X) $\vee$ animale(X)
2. $\neg$ cane(X) $\vee$ $\neg$ abbaia(X) $\vee$ spaventa(X)
3. $\neg$ animale (X) $\vee$ $\neg$ spaventa(X) $\vee$ $\neg$ amato(X)
4a. cane(c1) Skolem
4b. abbaia(c1) Skolem
GNeg: $\neg$ animale(X) $\vee$ amato(X)

Risoluzione:

5.: GNeg+1: $\neg$ cane(X) $\vee$ amato(X)
6.: 5 + 4a: amato(c1)
7.: 6 + 3: $\neg$ animale (c1) $\vee$ $\neg$ spaventa(c1)
8.: 7 + 2: $\neg$ cane(c1) $\vee$ $\neg$ abbaia(c1) $\vee$ $\neg$ animale(c1)
9.: 8 + 4b: $\neg$ cane(c1) $\vee$ $\neg$ animale(c1).
10.: 9+4a: $\neg$ animale(c1)
11: 10+1: $\neg$ cane (c1)
12: 11 + 4a: contraddizione!

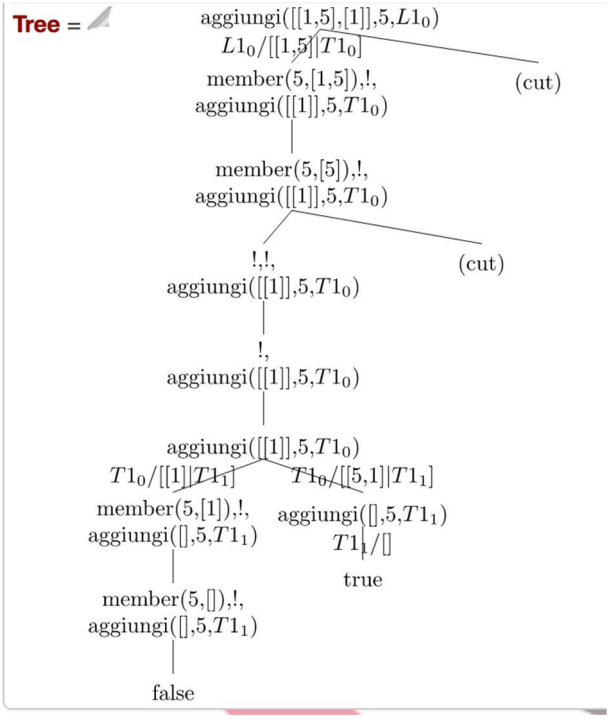
Esercizio 2





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

Esercizio 3

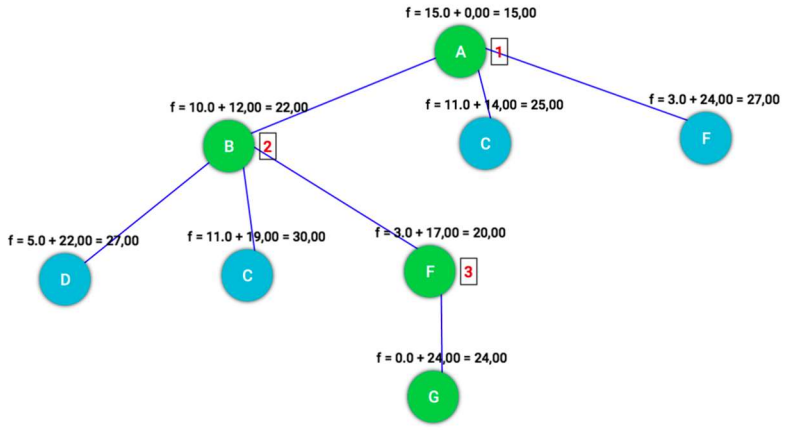


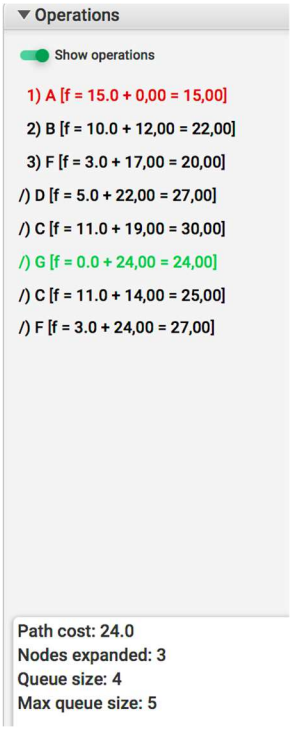
Esercizio 4

```
cresceSum([]).
cresceSum([_]).
cresceSum([H,H1|T]):- sum(H,A), sum(H1,B), A<B, cresceSum([H1|T]).

sum([X],X) :- !.
sum([H|T], N) :- sum(T,N1), N is N1+H.
```

Esercizio 5





L'euristica è ammissibile.
Il costo di cammino trovato da A* è 24 e il numero di nodi espansi per arrivare al goal G a partire dal nodo iniziale A è 3 (ABF) , non considerando il goal test sul goal G.

Esercizio 6
Vedi materiale del corso.