

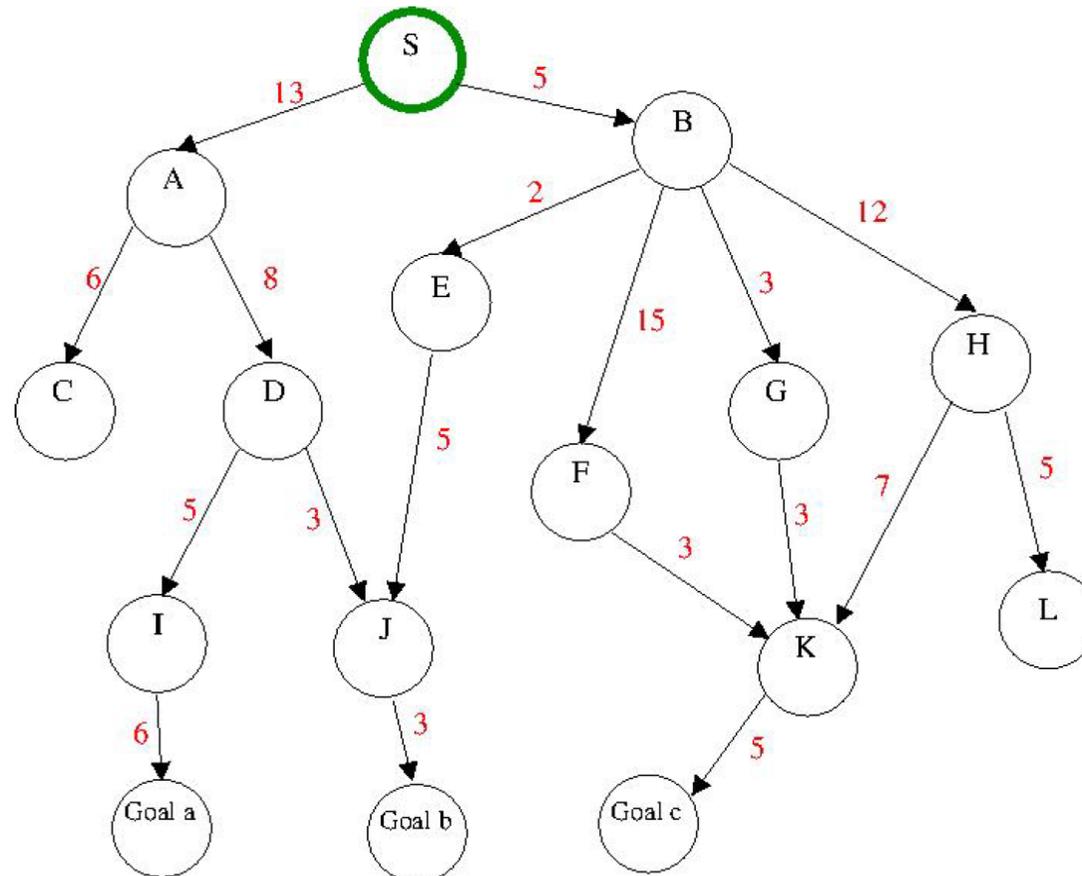
---

# Esercizi d'esame su strategie di ricerca, CSP e Giochi

Compiti d'esame 2017 e 2018

# Giugno 2018 – Strategia a costo uniforme

Si consideri il seguente grafo in cui gli archi sono annotati col costo e si indichi:  
(i) l'ordine dei nodi che sarebbero espansi nel caso di **ricerca a costo uniforme** per raggiungere uno dei tre nodi Goal,  
(ii) la strada che porta alla soluzione individuata.



# Giugno 2018 – soluzione

Nodi espansi (in grassetto il nodo scelto):

**S(0)**

A(13) **B(5)**

A(13) **E(7)** F(20) G(8) H(17)

A(13) J(12) F(20) **G(8)** H(17)

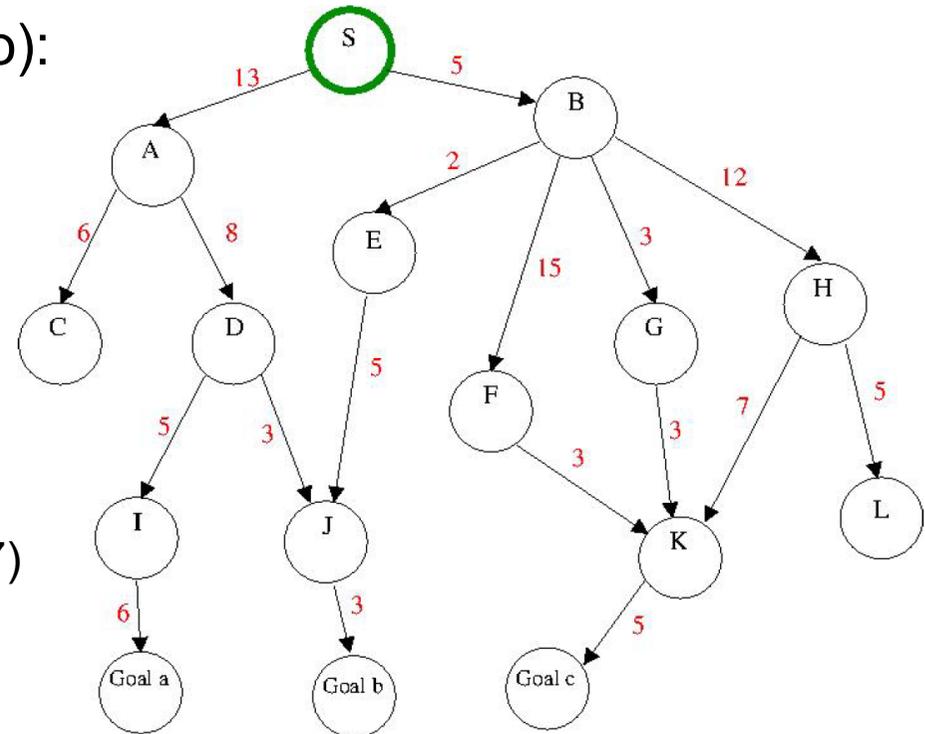
A(13) J(12) F(20) **K(11)** H(17)

A(13) **J(12)** F(20) Goalc(16) H(17)

**A(13)** Goalb(15) F(20) GoalC(16) H(17)

C(19) D(21) **Goalb(15)** F(20) GoalC(16) H(17)

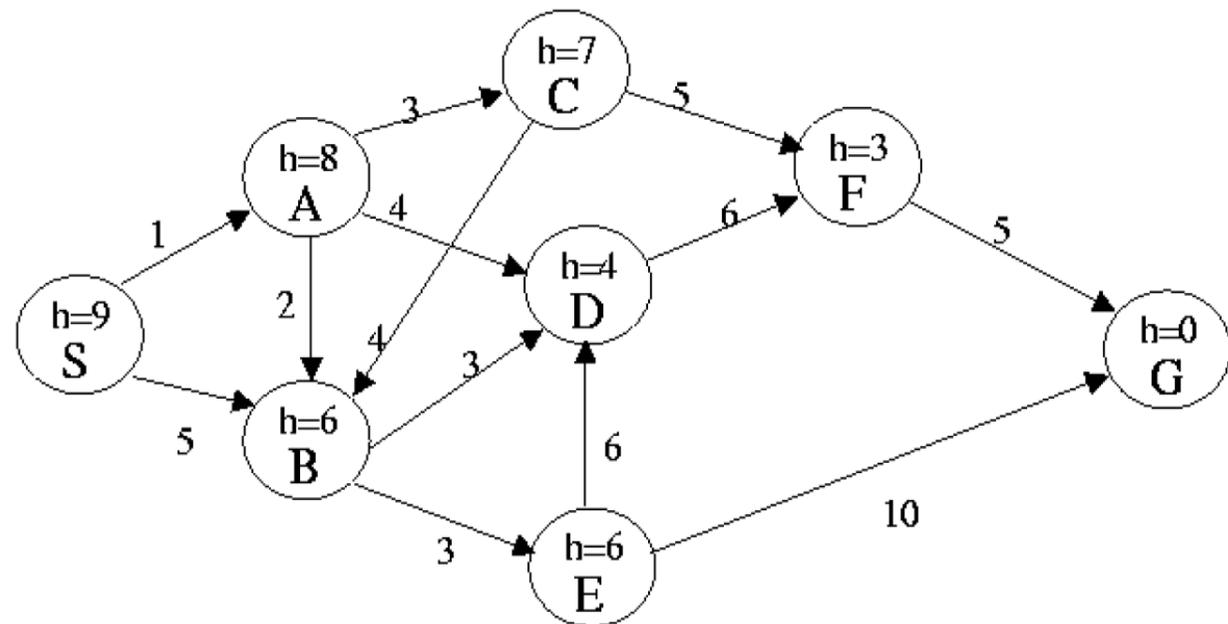
Percorso: S B E J Goalb.



# 11 Gennaio 2018 - Strategie

Dato il seguente grafo orientato, dove S è lo stato iniziale e G lo stato goal, il costo di ogni arco è indicato a fianco, e la stima euristica (h) della distanza di ogni nodo dall'obiettivo è riportata nel nodo, indicare l'ordine con cui sono visitati i nodi nel caso di ricerca:

- Depth-first (fino a 1° soluzione)
- Breadth-first
- Best-first (fino a 1° soluzione)
- A\*



Nel caso di scelte non-deterministiche tra nodi/stati, si scelga di muoversi sul nodo che è il primo secondo l'ordine alfabetico. Nota: l'espansione dei nodi già visitati può essere evitata senza conseguenze solo se il nuovo percorso verso lo stato ha un costo g (o funzione di valutazione totale f) superiore (o uguale) al costo dei percorsi già esplorati.

# 11 Gennaio 2018 - Strategie

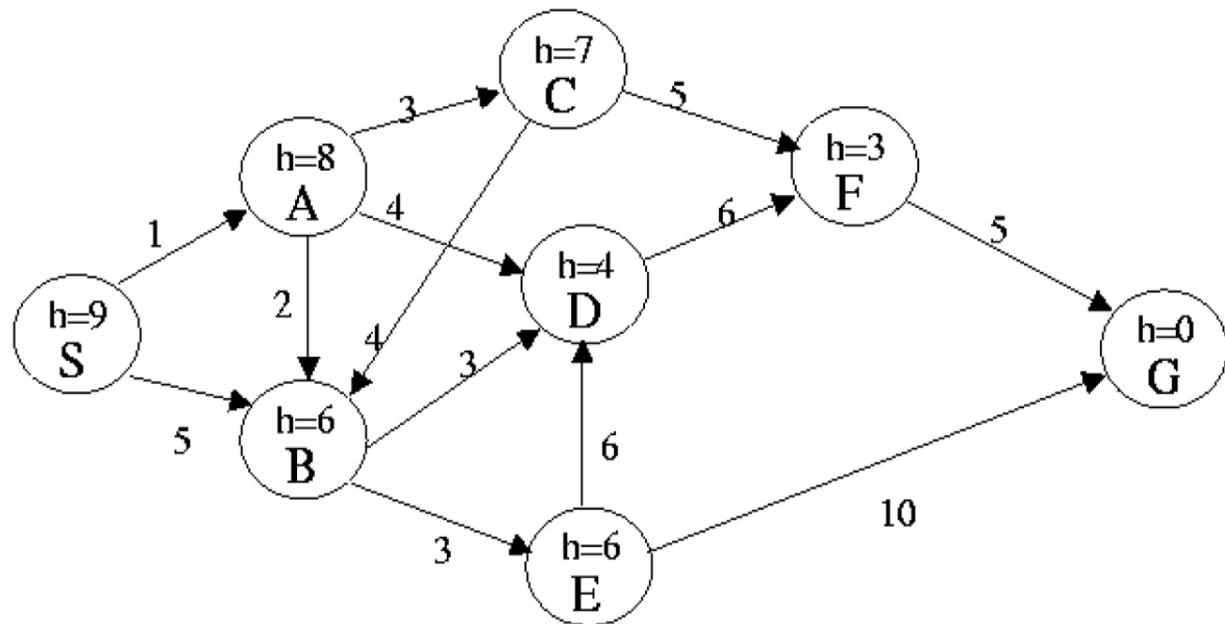
---

Depth-first (fino alla prima soluzione): S, A, B, D, F, G

Breadth-first: S, A, B, C, D, E, F, G

Best First Search: S, B, D, F, G

A\*: S, A, B, D, C, E, F, G

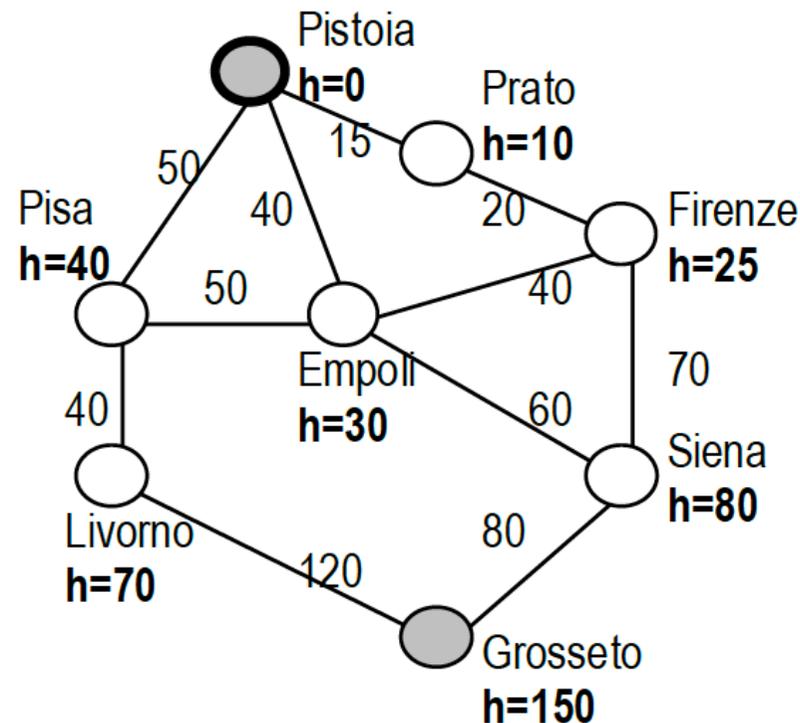


Disegnare gli alberi di ricerca ...

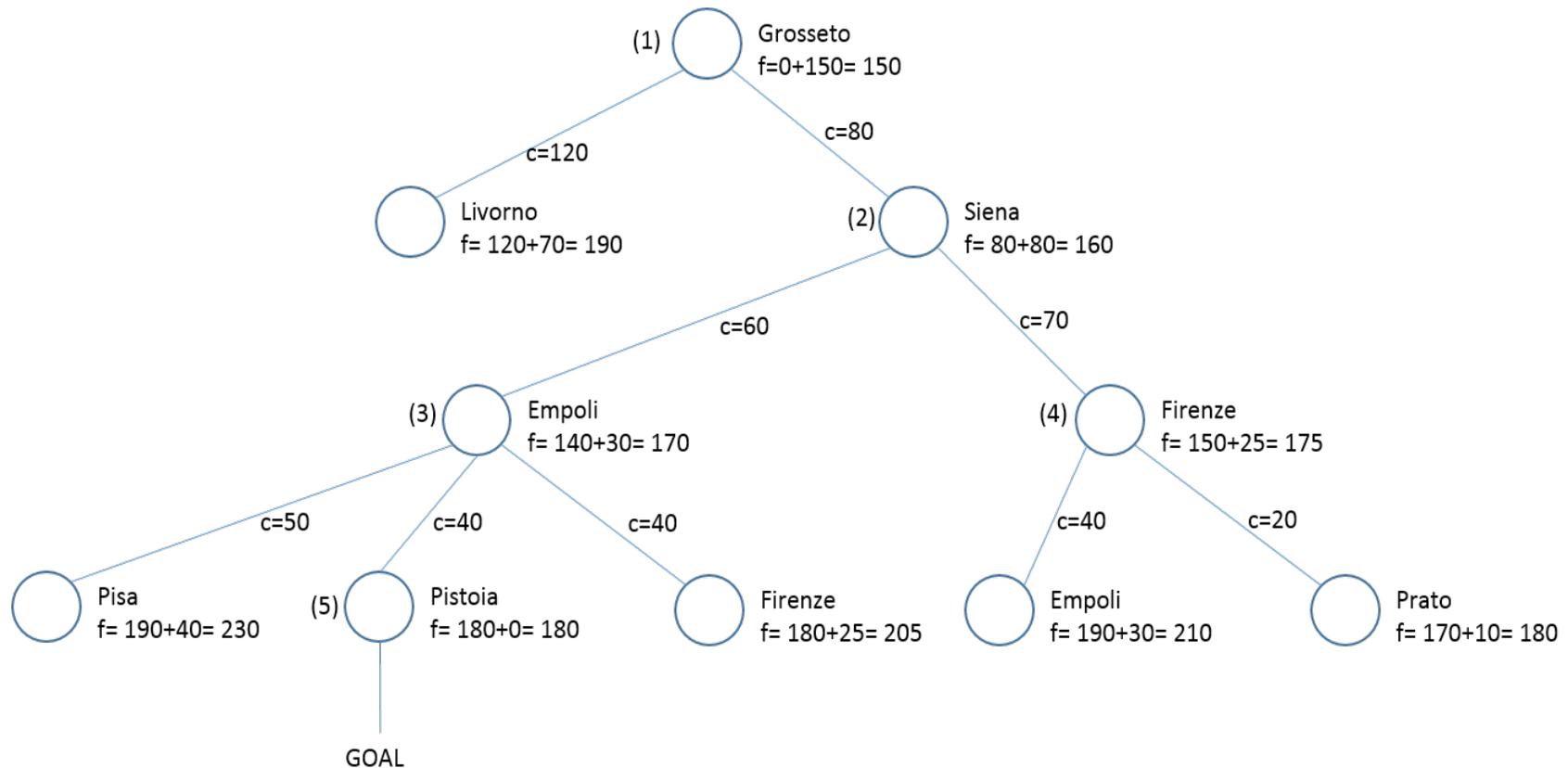
# Giugno 2017 – Strategia A\*

Dato il seguente grafo, si utilizzi l'**algoritmo A\*** per individuare il percorso più breve tra Grosseto e Pistoia. Accanto ad ogni città si trova la stima euristica della sua distanza da Pistoia. Accanto ad ogni strada si trova la lunghezza effettiva del tratto stradale. Si utilizzi A\* per trovare la soluzione e si mostri l'albero di ricerca espanso, dove i nodi sono considerati da Ovest a Est (cioè, i nodi più a Ovest saranno i primi a sinistra nell'albero di ricerca, e viceversa) e nel dubbio A\* espande sempre il nodo più a sinistra.

**L'euristica indicata è ammissibile?**



# Giugno 2017 – soluzione



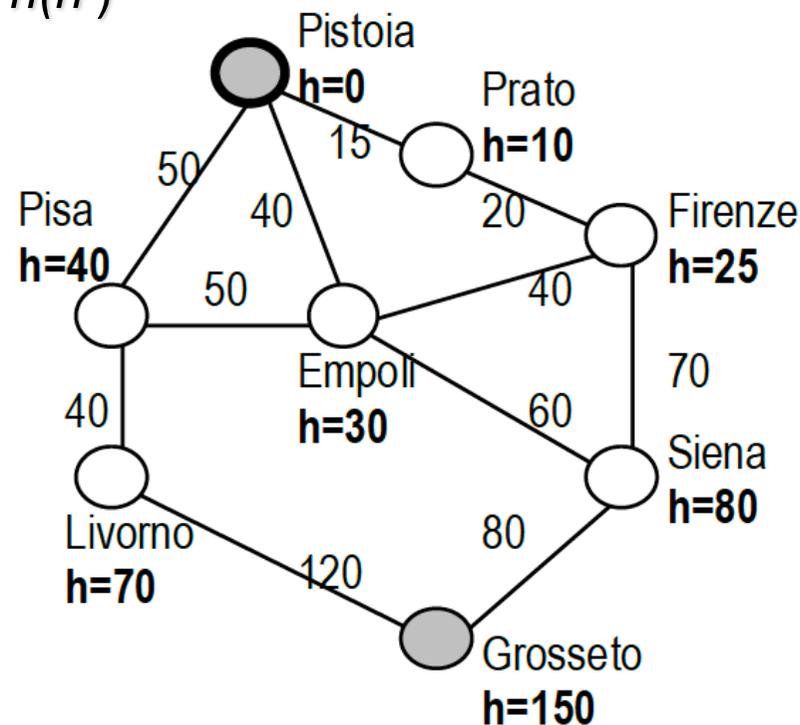
**Risposta:** La stima euristica è ammissibile perché per ogni nodo è minore o uguale del costo effettivo per raggiungere il Goal a partire da tale nodo.

# Giugno 2017 – Strategia A\*

- L'euristica indicata è anche consistente (consistenza o monotonicità)?
- Definizione: una euristica è **consistente** se per ogni nodo  $n$ , ogni successore  $n'$  di  $n$  generato da ogni azione  $a$ ,

$h(n) = 0$  se lo stato corrispondente coincide con il goal

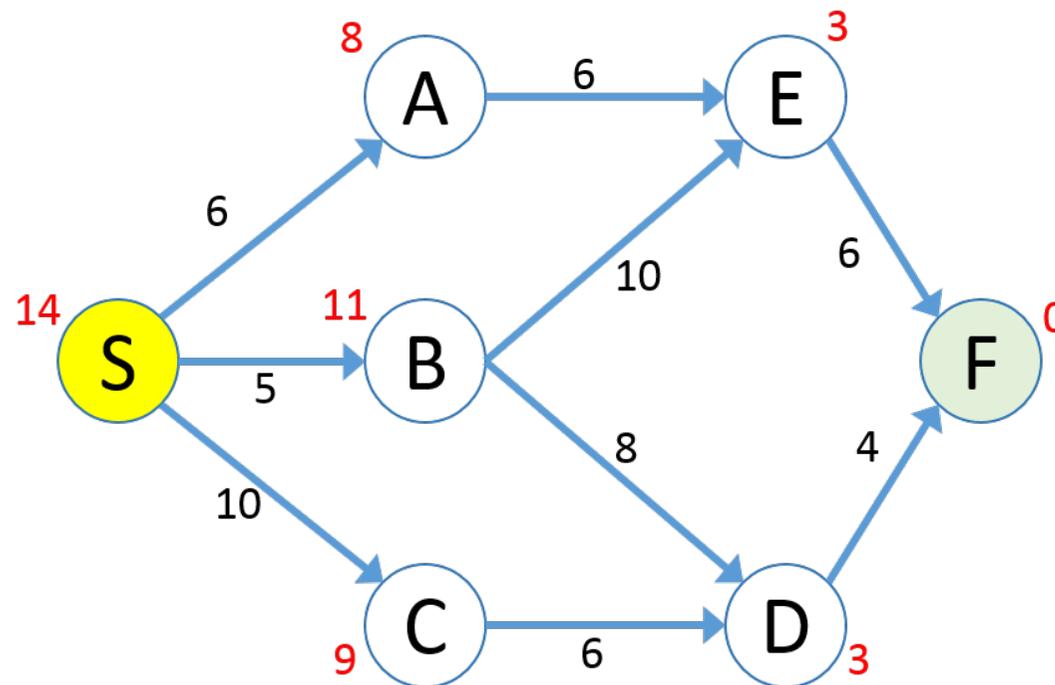
$$h(n) \leq c(n,a,n') + h(n')$$



Se consistente, ogni nodo è raggiunto per primo con la strada migliore

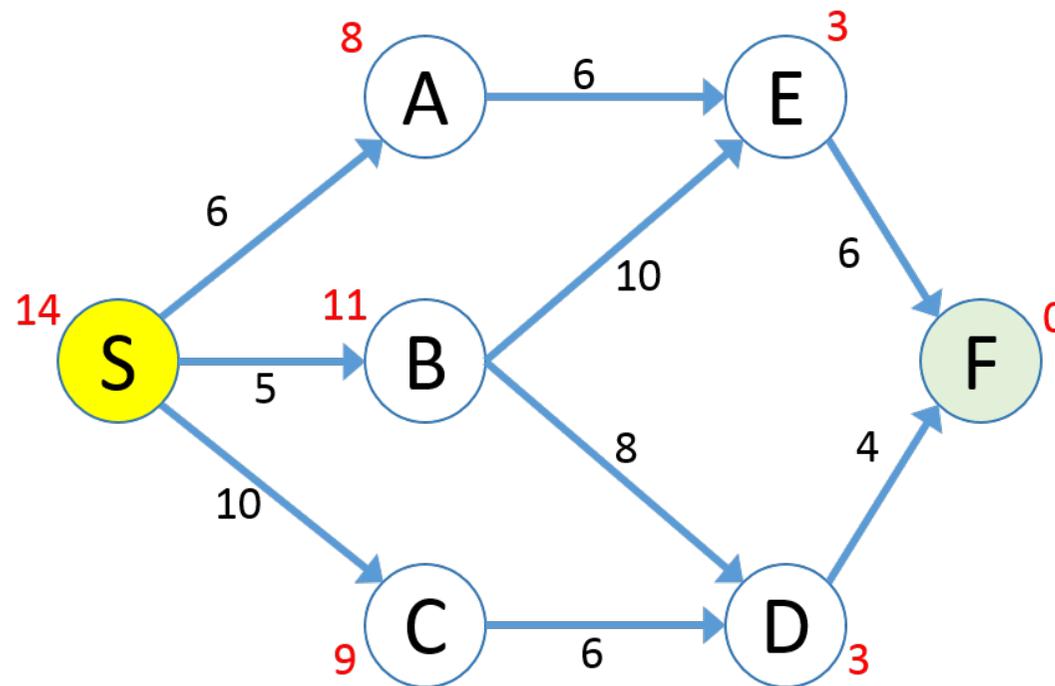
## Settembre 2017 – Strategia A\*

Si consideri il seguente grafo, dove ogni arco è etichettato con la sua lunghezza (in nero). Si vuole andare dal nodo S al nodo F, determinando con l'algoritmo A\* il percorso. Ogni nodo è etichettato con la stima della sua distanza dal nodo F (in rosso). Tale stima è ammissibile?



## Settembre 2017 – Strategia A\*

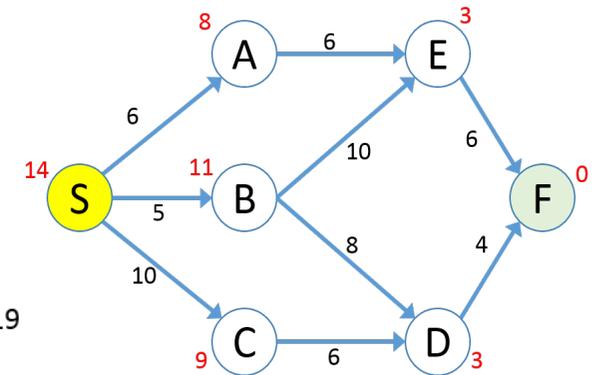
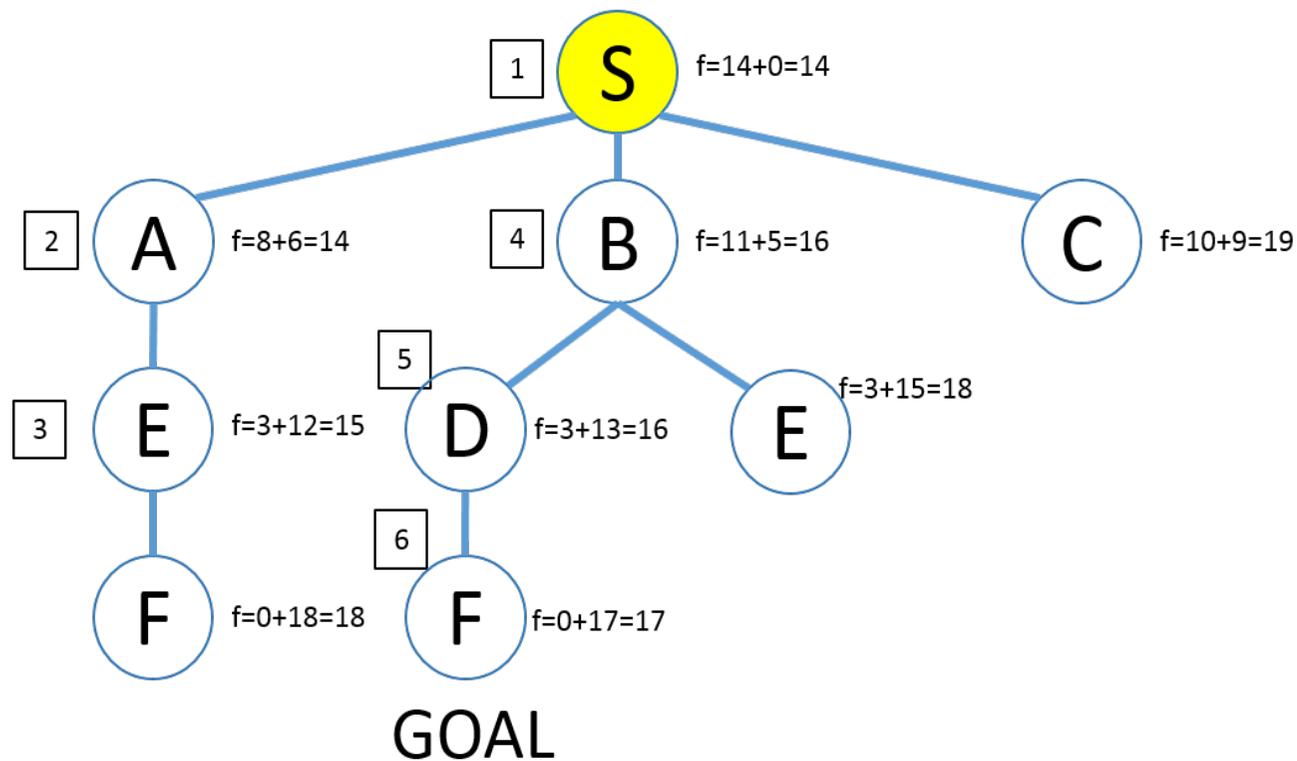
Si consideri il seguente grafo, dove ogni arco è etichettato con la sua lunghezza (in nero). Si vuole andare dal nodo S al nodo F, determinando con l'algoritmo A\* il percorso. Ogni nodo è etichettato con la stima della sua distanza dal nodo F (in rosso). Tale stima è ammissibile?



**Risposta:** La stima euristica è ammissibile perché per ogni nodo è minore o uguale del costo effettivo per raggiungere il Goal a partire da tale nodo.

# Settembre 2017 – Strategia A\*

Mostrare l'albero di ricerca sviluppato con A\* applicando la ricerca su albero senza eliminazione dei nodi ripetuti, indicando l'ordine con cui sono espansi i nodi (indicare tale ordine con un numero tra parentesi quadre a fianco di ogni nodo espanso dell'albero) fino al raggiungimento della soluzione.



# Constraint Satisfaction Problems

---

## Luglio 2017 - CSP

---

- **Formulare il seguente problema** come CSP, individuando variabili, domini e vincoli unari e binari:
  - “Alice, Barbara, Carla e Daria sono 4 bambine di età (espressa in anni) compresa fra 1 e 10 anni.
  - Esse possiedono 4 cagnolini, rispettivamente Rococò, Sibilo, Tocco e Uguccione, anch’essi di età (espressa in anni) compresa fra 1 e 10 anni.
  - L’età di ciascuna padroncina è il doppio di quella del proprio cane.
  - Alice ha l’età del cane di Barbara.”
- Si applichi poi **l’algoritmo AC3**, mostrando se e come sono ridotti i domini delle variabili.
- E’ possibile identificare già una soluzione dopo l’applicazione di tale algoritmo?

# Luglio 2017 – Modello CSP

---

- “Alice, Barbara, Carla e Daria sono 4 bambine di età (espressa in anni) compresa fra 1 e 10 anni.
- Esse possiedono 4 cagnolini, rispettivamente Rococò, Sibilo, Tocco e Ugucione, anch’essi di età (espressa in anni) compresa fra 1 e 10 anni.

- Variabili e Domini:

A, B, C, D, per le bambine;      R, S, T, U per i cani.

$DA = DB = DC = DD = [1..10]$        $DR = DS = DT = DU = [1..10]$

- L'età di ciascuna padroncina è il doppio di quella del proprio cane.
- Alice ha l'età del cane di Barbara.”

- Vincoli binari:

$A = 2 * R$        $B = 2 * S$

$C = 2 * T$        $D = 2 * U$

$A = S$

# Luglio 2017 – AC3 su CSP

AC-3	A	B	C	D	R	S	T	U
	1..10	1..10	1..10	1..10	1..10	1..10	1..10	1..10
Var1=2*Var2	2,4,6,8,10	2,4,6,8,10	2,4,6,8,10	2,4,6,8,10	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5
Vincolo A=S	2,4					2,4		
Var1=2*Var2		4,8			1,2			

Non è possibile determinare una soluzione direttamente (i domini delle variabili contengono più valori). Occorrerebbe quindi applicare ricerca (labeling) e un algoritmo di propagazione (eventualmente ancora AC-3, ma dopo labeling)

# Luglio 2018 – ... con modellazione CSP

---

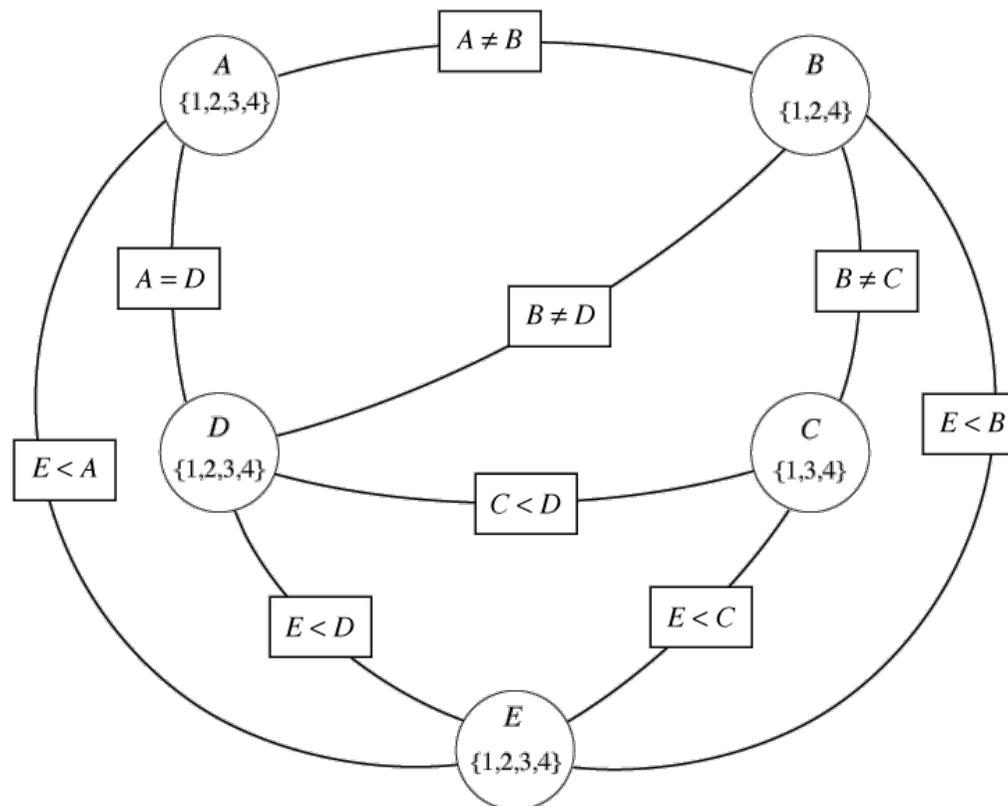
- In una fabbrica, su una serie di macchine in parte condivise, si vogliono svolgere alcune **attività denominate A, B, C, D ed E**. Ogni attività può essere **eseguita nell'intervallo di tempo numero 1, 2, 3, o 4**. Gli intervalli sono distinti, successivi e adiacenti nel tempo, e non si intersecano (per questo li mappiamo negli interi da 1 a 4).
- Si hanno i seguenti **vincoli**:
  - B non può essere fatta nell'intervallo 3
  - C non può essere fatta nell'intervallo 2
  - A e B richiedono la stessa macchina e non si possono svolgere nello stesso intervallo tempo,
  - B e C richiedono la stessa macchina e non si possono svolgere nello stesso intervallo tempo,
  - L'attività C deve essere fatta prima della D
  - L'attività A e D devono essere fatte contemporaneamente
  - L'attività E deve essere fatta prima di tutte le altre
  - B e D richiedono la stessa macchina e non si possono svolgere nello stesso intervallo tempo.
- Si modelli il problema come un CSP, determinando variabili e domini.
- Si disegni poi il corrispondente Constraint Graph e si applichi la Nodwe-consistency.
- Successivamente applichi l'Arc-Consistency al Constraint graph in esame, e si mostrino i domini ridotti dopo l'applicazione.
- Si determini poi la soluzione, se esiste.

# Luglio 2018 – modello CSP

---

- A, B, C, D, E :: [1..4]
  - B non può essere fatta nell'intervallo 3
  - C non può essere fatta nell'intervallo 2
  - A e B richiedono la stessa macchina e non si possono svolgere nello stesso intervallo tempo,
  - B e C richiedono la stessa macchina e non si possono svolgere nello stesso intervallo tempo,
  - L'attività C deve essere fatta prima della D
  - L'attività A e D devono essere fatte contemporaneamente
  - L'attività E deve essere fatta prima di tutte le altre
  - B e D richiedono la stessa macchina e non si possono svolgere nello stesso intervallo tempo
- Vincoli:  
{(B≠3), (C≠2), (A≠B), ( B≠C), (C<D), (A=D),  
(E<A), (E<B), (E<C), (E<D), ( B≠D)}

# Luglio 2018 – grafo (già node consistent) e AC



Supponiamo che l'arco  $\langle D, C < D \rangle$  sia considerato per primo. L'arco non è coerente perché  $D = 1$  non è coerente con alcun valore nel dominio di C, quindi 1 viene eliminato dal dominio di D che diventa  $\{2,3,4\}$ . Supponiamo ora di considerare l'arco  $\langle C, E < C \rangle$  Il dominio di C viene ridotto a  $\{3,4\}$  e l'arco  $\langle D, C < D \rangle$  dovrà essere riconsiderato.

Supponiamo che l'arco  $\langle D, C < D \rangle$  sia il prossimo; quindi il dominio di D viene ulteriormente ridotto al singleton  $\{4\}$ . L'arco  $\langle C, C < D \rangle$  riduce il dominio di C a  $\{3\}$ . L'arco  $\langle A, A = D \rangle$  riduce il dominio di A a  $\{4\}$ . L'elaborazione di  $\langle B, B \neq D \rangle$  riduce il dominio di B a  $\{1,2\}$ .

Quindi l'arco  $\langle B, E < B \rangle$  riduce il dominio di B a  $\{2\}$ . Infine, l'arco  $\langle E, E < B \rangle$  riduce il dominio di E a  $\{1\}$ .

Tutti gli archi rimasti nella coda sono soddisfatti e quindi l'algoritmo arriva a uno stato di quiescenza e termina. Vengono quindi restituiti i domini delle variabili opportunamente ridotti.

In questo caso, i domini hanno tutti cardinalità 1 e esiste un'unica soluzione:  $A = 4, B = 2, C = 3, D = 4, E = 1$  che si può verificare rispetta tutti i vincoli.

# 11 Gennaio 2018 – CSP

---

- Dopo avere brevemente introdotto le tecniche di propagazione di Forward Cheking (FC) e Partial (PLA) e Full (FLA) Look Ahead se ne mostri l'esecuzione su questo esempio mostrando la riduzione dei domini delle restanti variabile quando viene istanziata la variabile X1 a 4 (considerare le variabili secondo l'ordine del loro pedice):
  - $X1, X2, X3 :: [1, 2, 3, 4, 5, 6]$
  - $X1 > X2$
  - $X2 > X3$
  - $X1 > X3$
  - $X1 = 4$

# 11 Gennaio 2018 – CSP soluzione

---

Descrizione FC, PLA e FLA .... (vedi slide del corso)

Domini:  $X1, X2, X3 :: [1, 2, 3, 4, 5, 6]$

Vincoli:  $X1 > X2$   $X2 > X3$   $X1 > X3$   $X1 = 4$

	X1	X2	X3
<b>FC</b>	[1,2,3,4,5,6]	[1,2,3,4,5,6]	[1,2,3,4,5,6]
<b>X1=4, X1&gt;X2, X1&gt;X3</b>	4	[1,2,3]	[1,2,3]
<b>PLA</b>			
<b>X1=4, X1&gt;X2, X1&gt;X3</b>	4	[1,2,3]	[1,2,3]
<b>X2&gt;X3</b>		[2,3]	[1,2,3]
<b>FLA</b>			
<b>X1=4, X1&gt;X2, X1&gt;X3</b>	4	[1,2,3]	[1,2,3]
<b>X2&gt;X3</b>		[2,3]	[1,2,3]
<b>X3&lt;X1</b>		[2,3]	[1,2,3]
<b>X3&lt;X2</b>		[2,3]	[1,2]

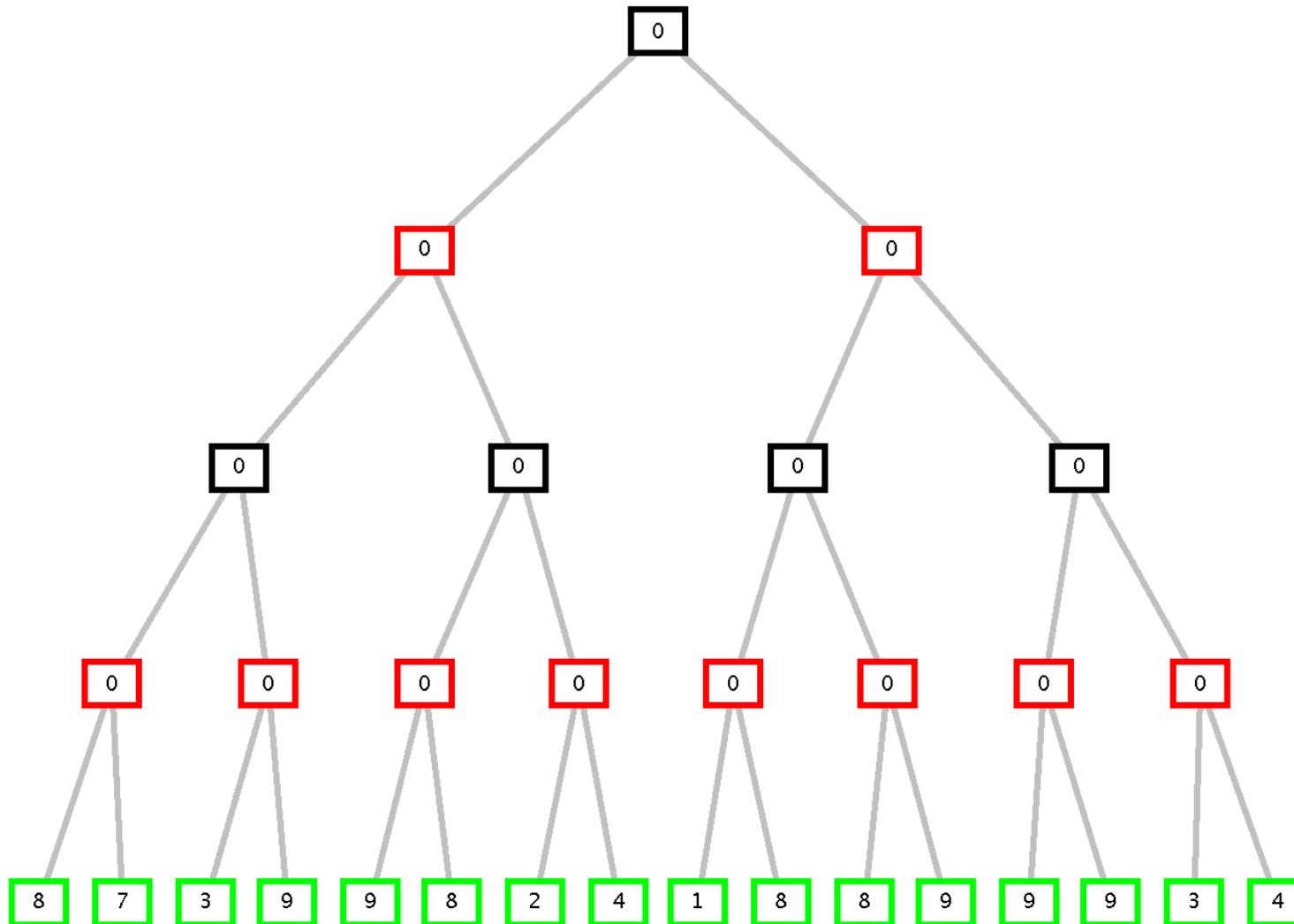
# Giochi – ricerca con avversario

---

# Giugno 2017

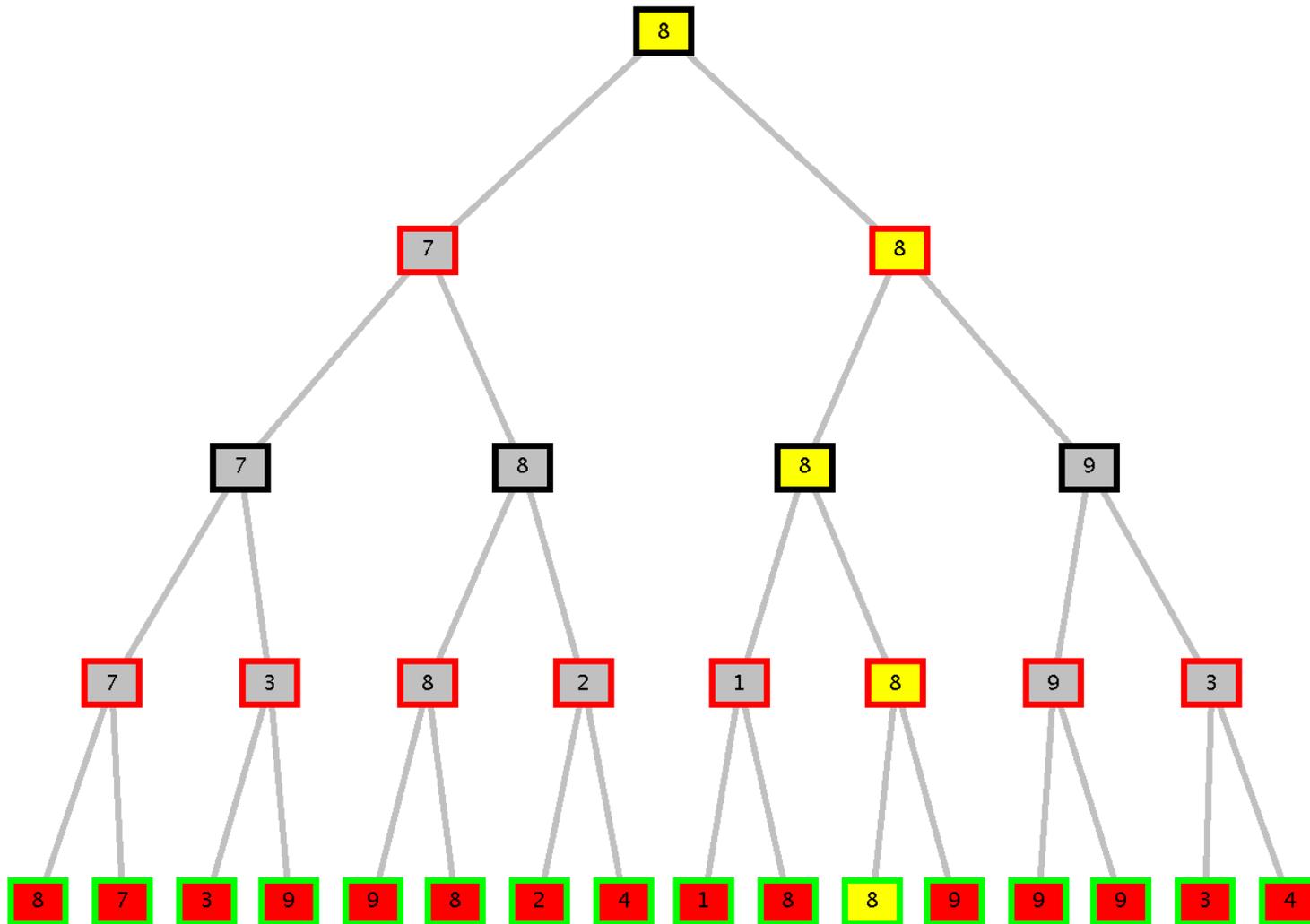
Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore (*MAX*). Si mostri come l'algoritmo *min-max* e l'algoritmo *alfa-beta* risolvono il problema e la mossa selezionata dal giocatore.

---



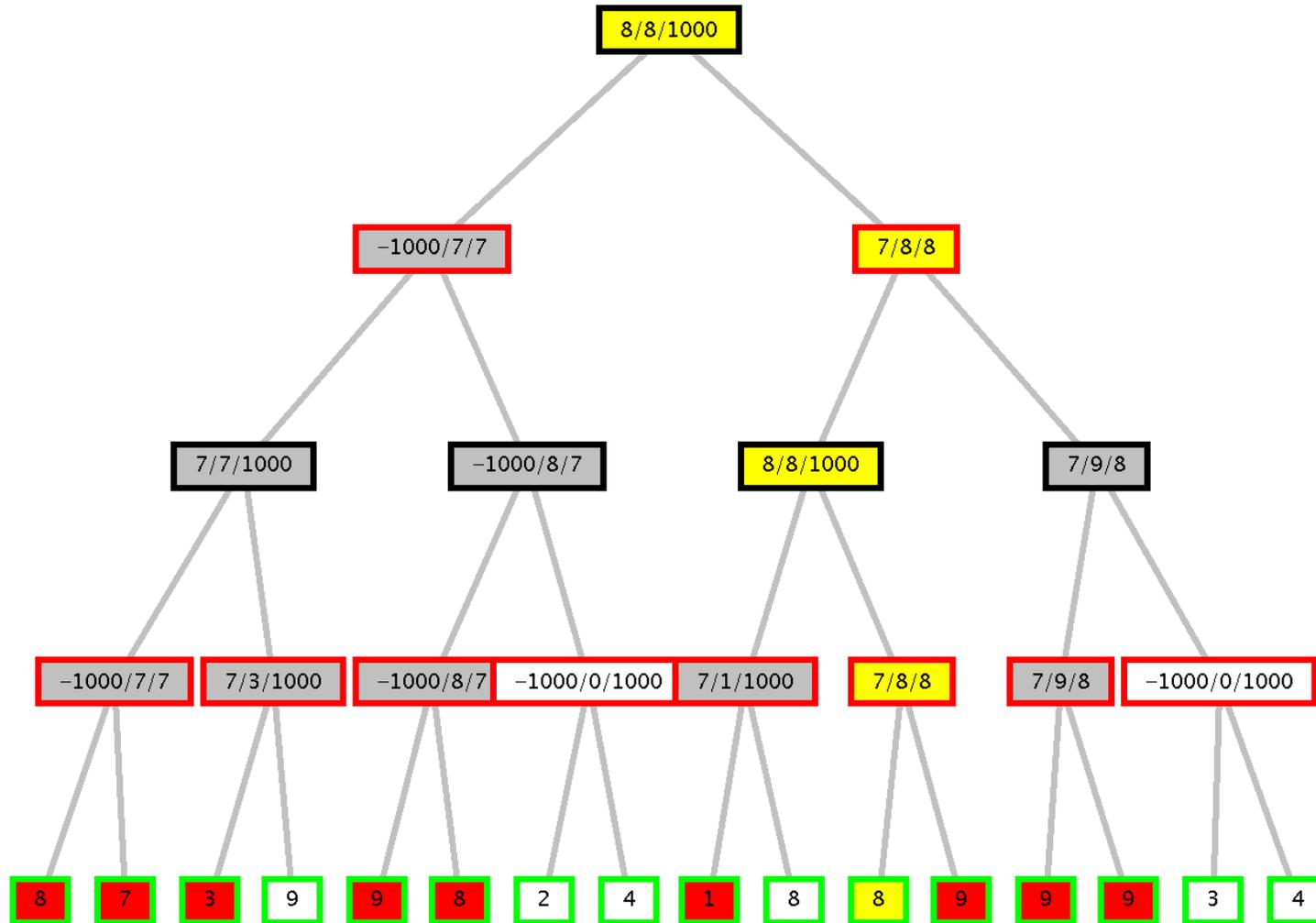
# Giugno 2017 – min-max

---



# Giugno 2017 – alfa-beta

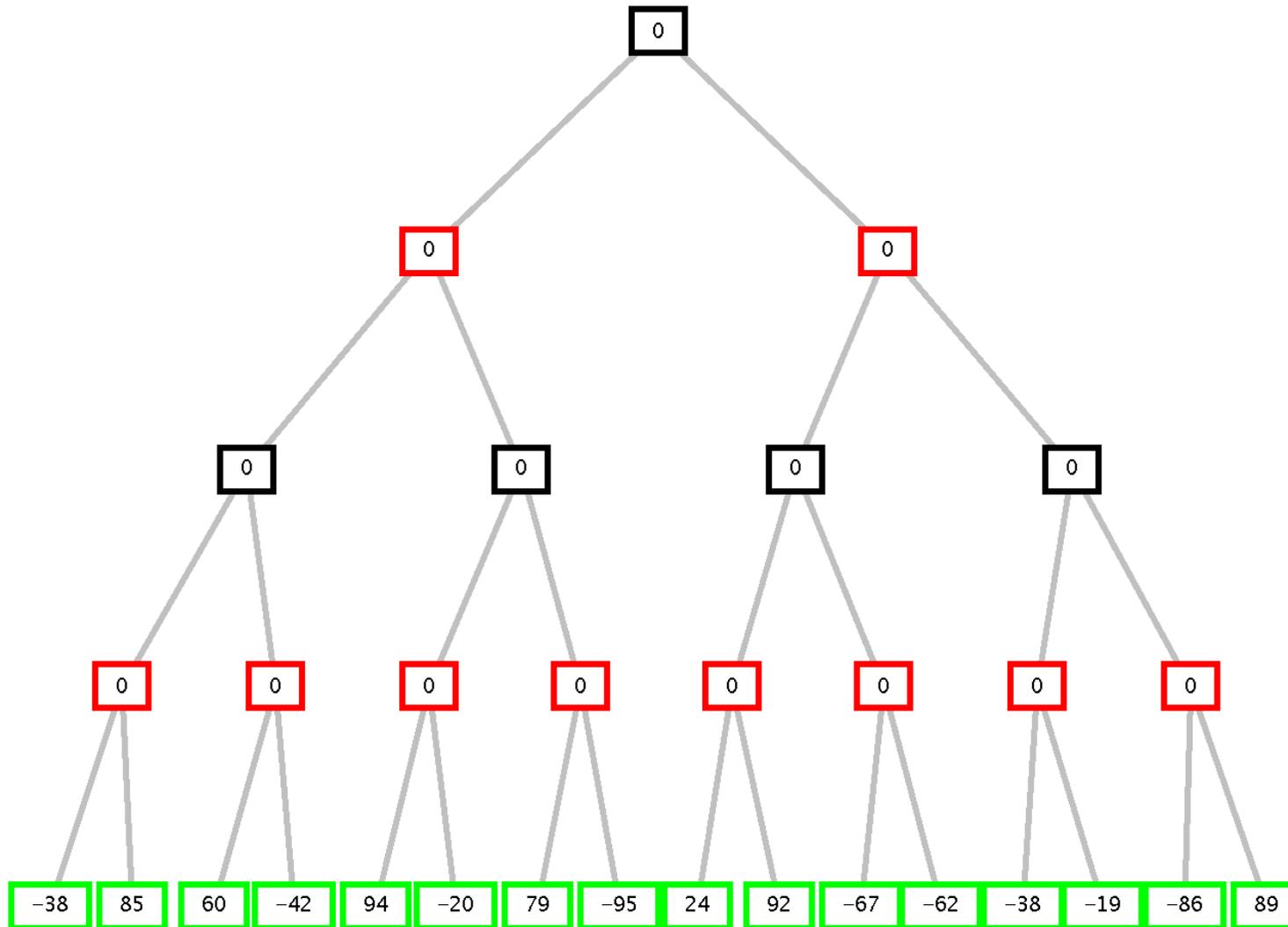
---



# Settembre 2017

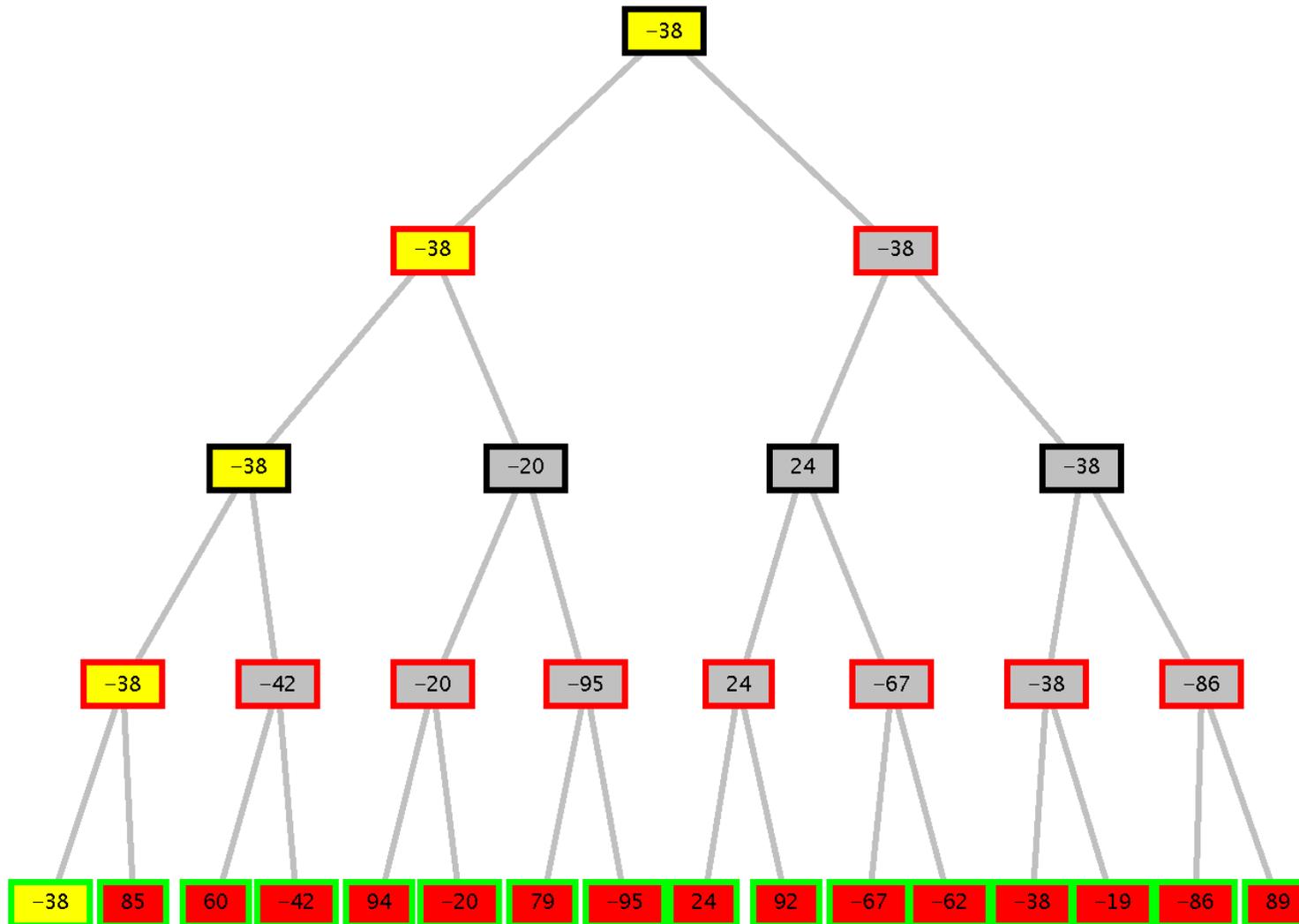
Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore (*MAX*). Si mostri come l'algoritmo *min-max* e l'algoritmo *alfa-beta* risolvono il problema e la mossa selezionata dal giocatore.

---

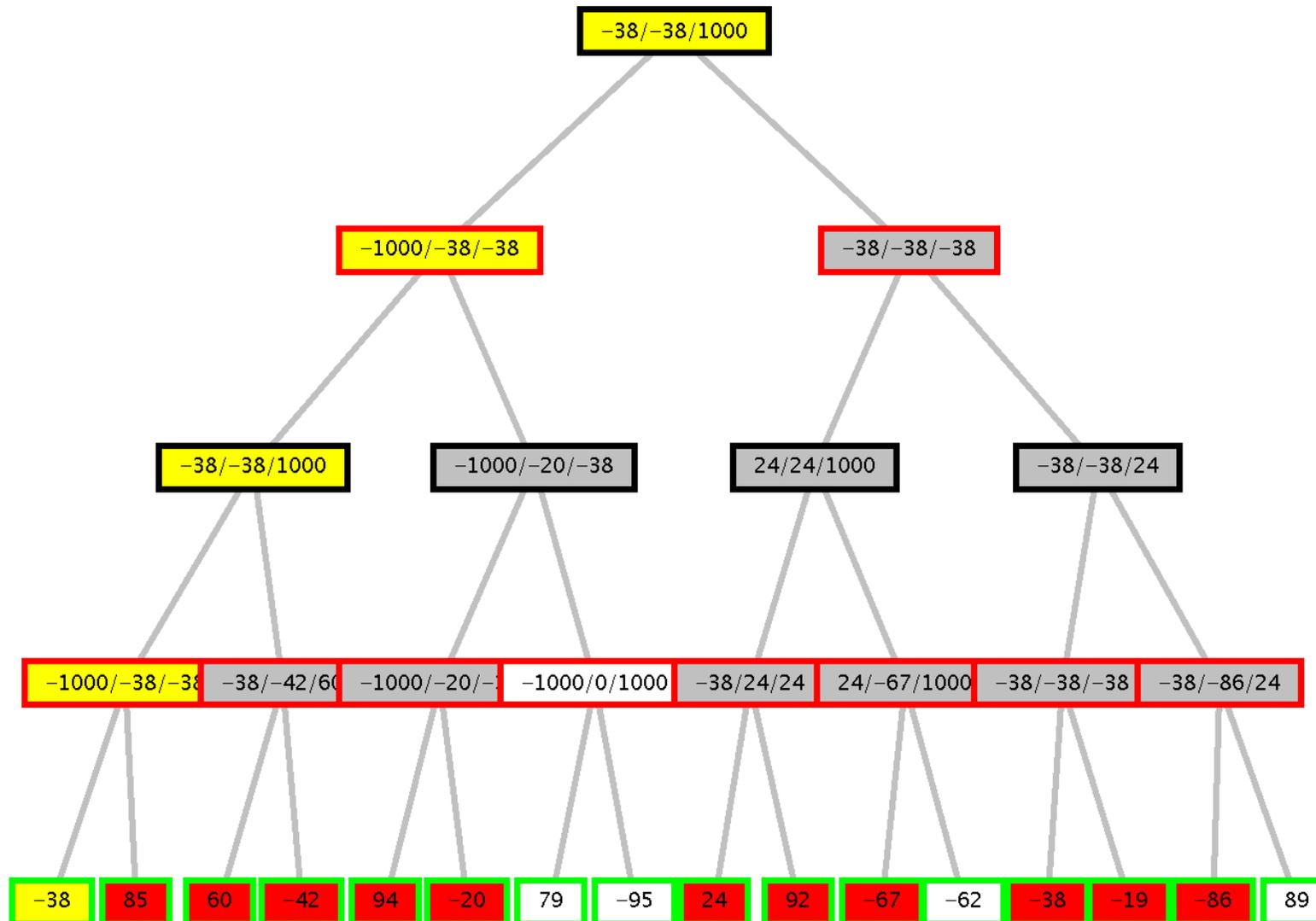


# Settembre 2017 – min-max

---



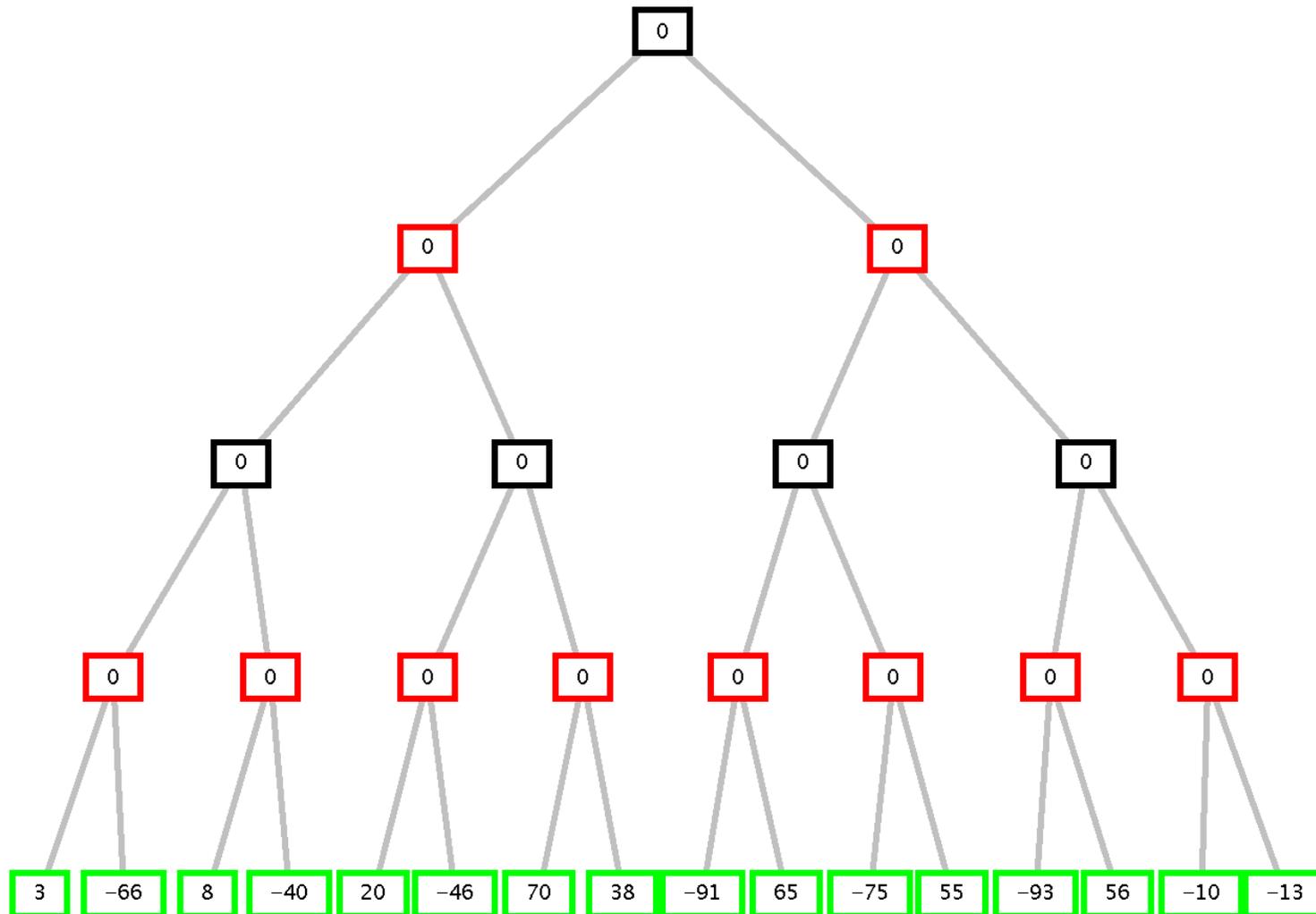
# Settembre 2017 – alfa-beta



# Luglio 2017

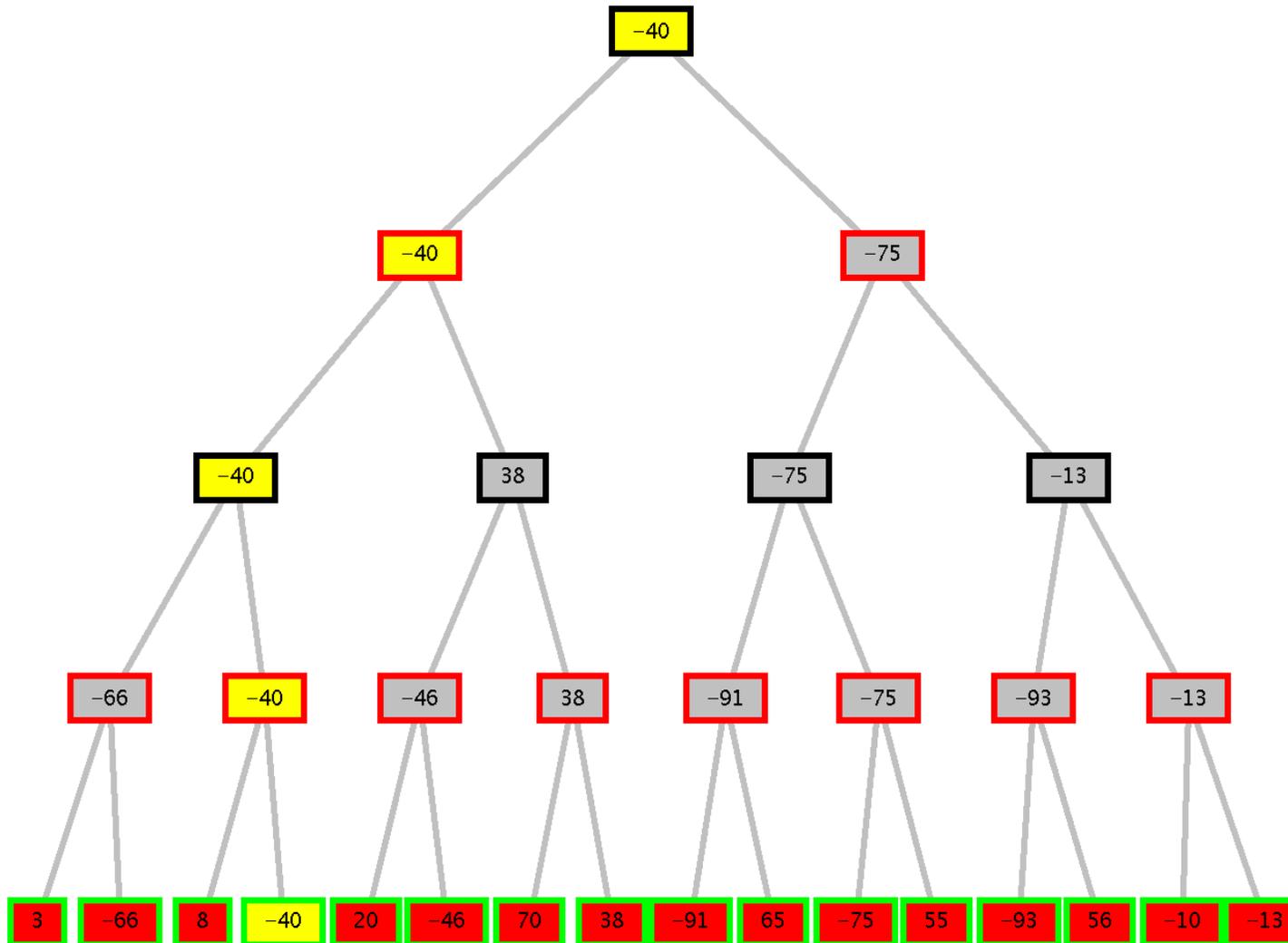
Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore (*MAX*). Si mostri come l'algoritmo *min-max* e l'algoritmo *alfa-beta* risolvono il problema e la mossa selezionata dal giocatore.

---



# Luglio 2017 – min-max

---



# Luglio 2017 – alfa-beta

