Strategie di ricerca

- *Scopo:*
  1. Esercizi sulle strategie di ricerca
  2. Esercizi sui giochi, alberi min-max e tagli alfa-beta

Ricordiamoci che…

- **Una funzione euristica *h(n)* è detta *ammissibile* se *non sbaglia mai per eccesso* la stima del costo per arrivare all'obbiettivo**

- **Una funzione euristica *h(n)* è detta *consistente* (o *monotona*) se, per ogni nodo *n* ed ogni successore *n'* generato da *n* applicando l'azione *a*, vale:**

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

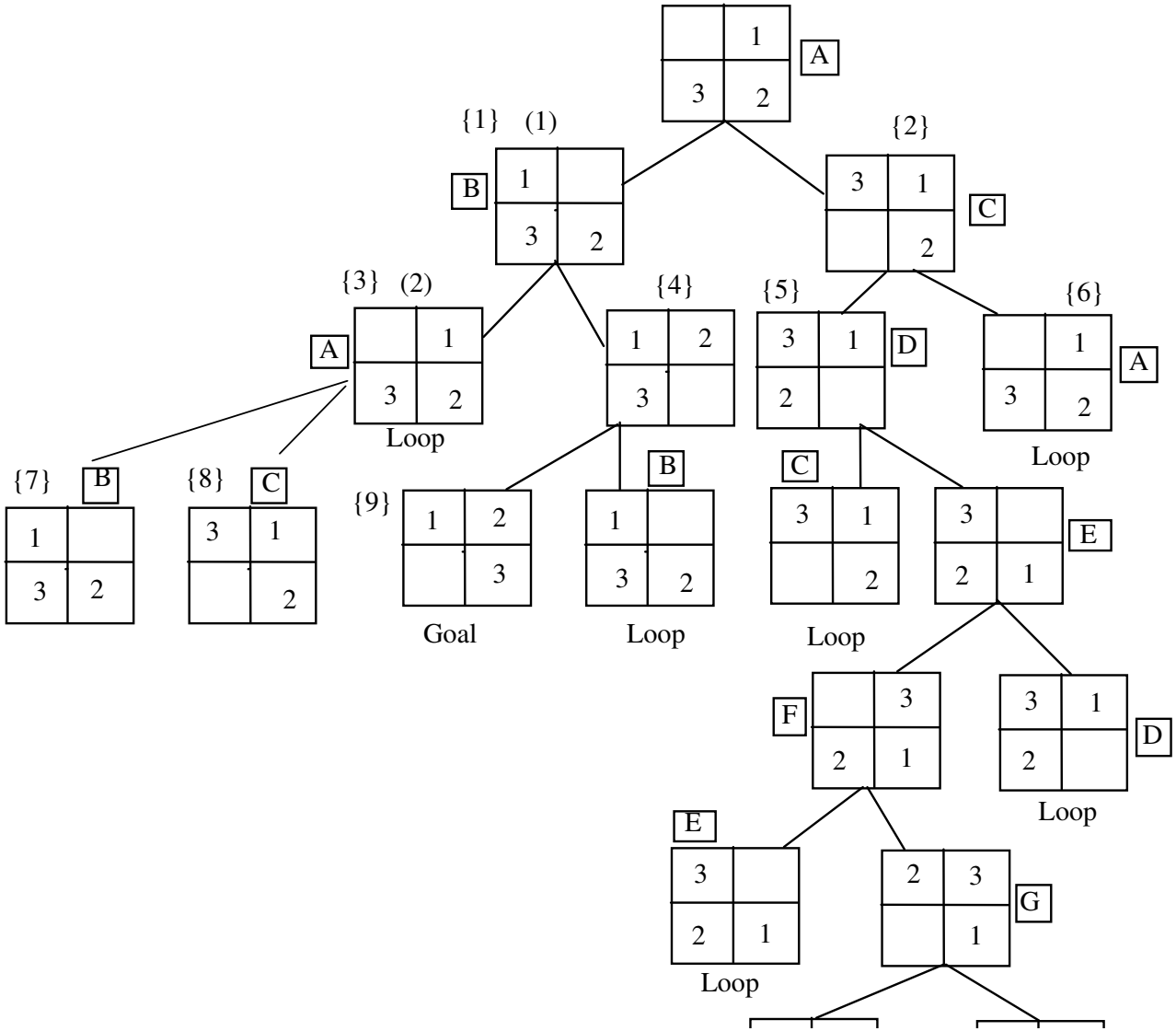
Esercizio – strategie di ricerca

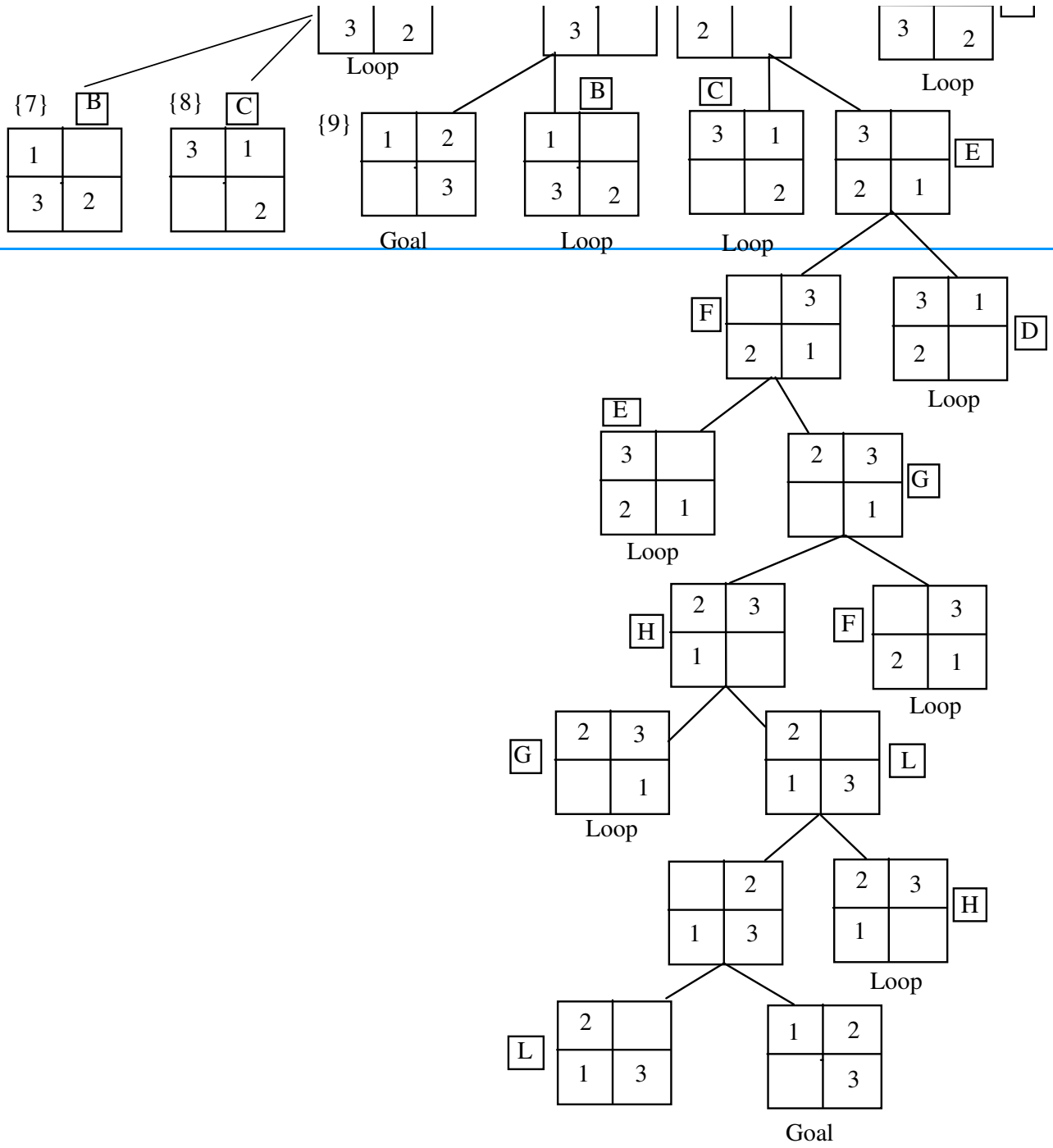
Si consideri un gioco del filetto di dimensione 3 con i seguenti stati iniziali e finali:

- **Si assuma che si preferisca prima muovere il tassello bianco a destra, poi a sinistra, poi in alto ed infine in basso e si mostri lo spazio di ricerca per il problema. Si mostri poi come viene esplorato tale spazio nei casi di depth-first e breadth-first.**

- **Si mostri poi come cambia lo spazio degli stati se non consentiamo che una mossa torni a uno stato precedente.**

Soluzione – strategie di ricerca



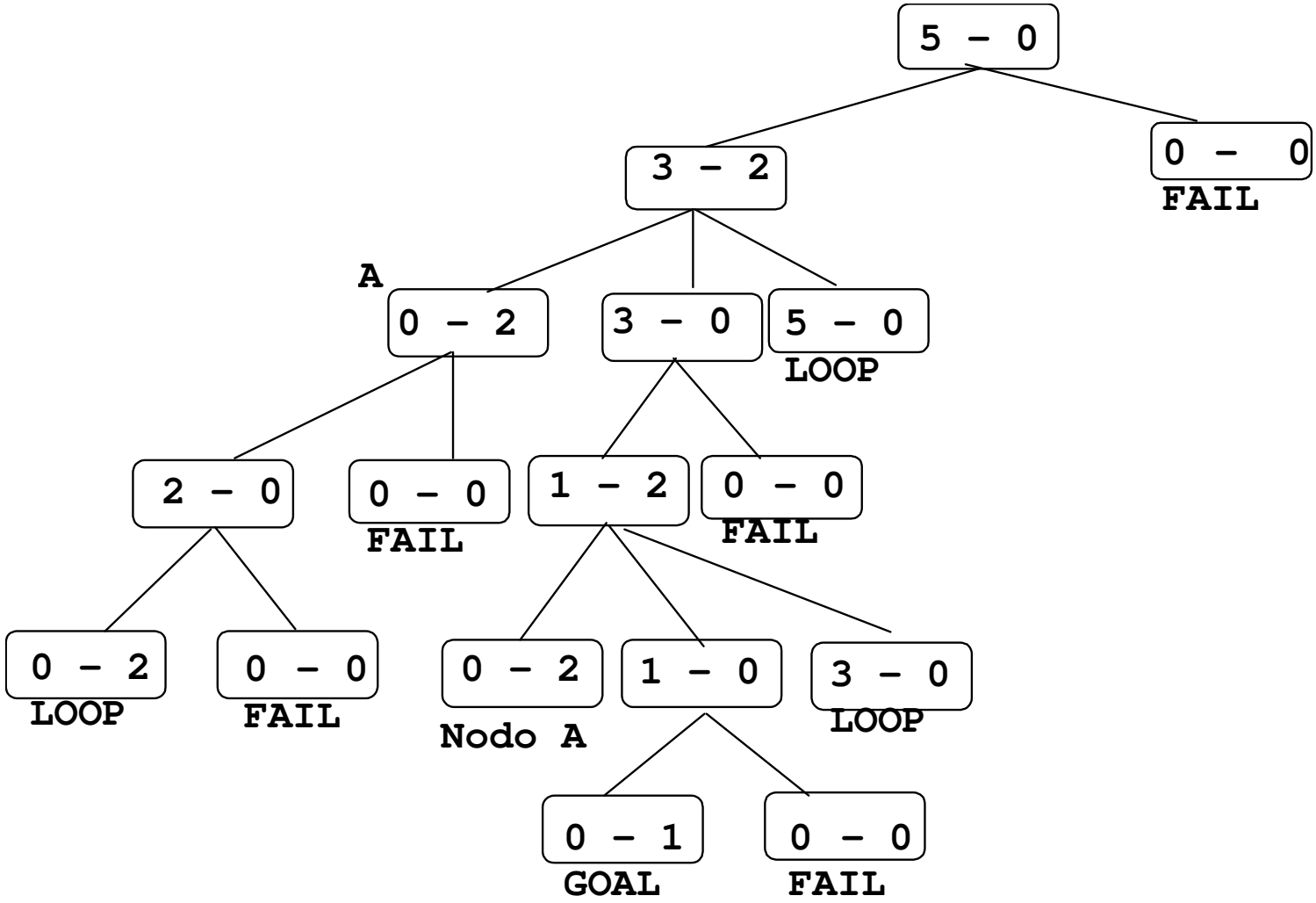

{7}
B
{8}
C
{9}
B
C
E
F
D
E
G
H
F
G
L
H
L
Loop
Goal
Loop
Loop
Loop
Loop
Loop
Loop
Loop
Goal

Esercizio – strategie di ricerca

Si consideri il seguente problema. Si hanno due contenitori per liquidi, uno da cinque litri completamente pieno d'acqua e uno da due litri vuoto.

- **Vogliamo ottenere precisamente un litro d'acqua nel contenitore da due litri.**
- **Possiamo trasferire acqua da un contenitore all'altro, o buttarla, ma non ottenerne di nuova.**
- **Si mostri tutto lo spazio degli stati per risolvere il problema. Si indichi, inoltre, dopo quanti passi si giungerebbe alla soluzione se si utilizzasse una strategia breadth-first e depth-first.**

Soluzione – strategie di ricerca

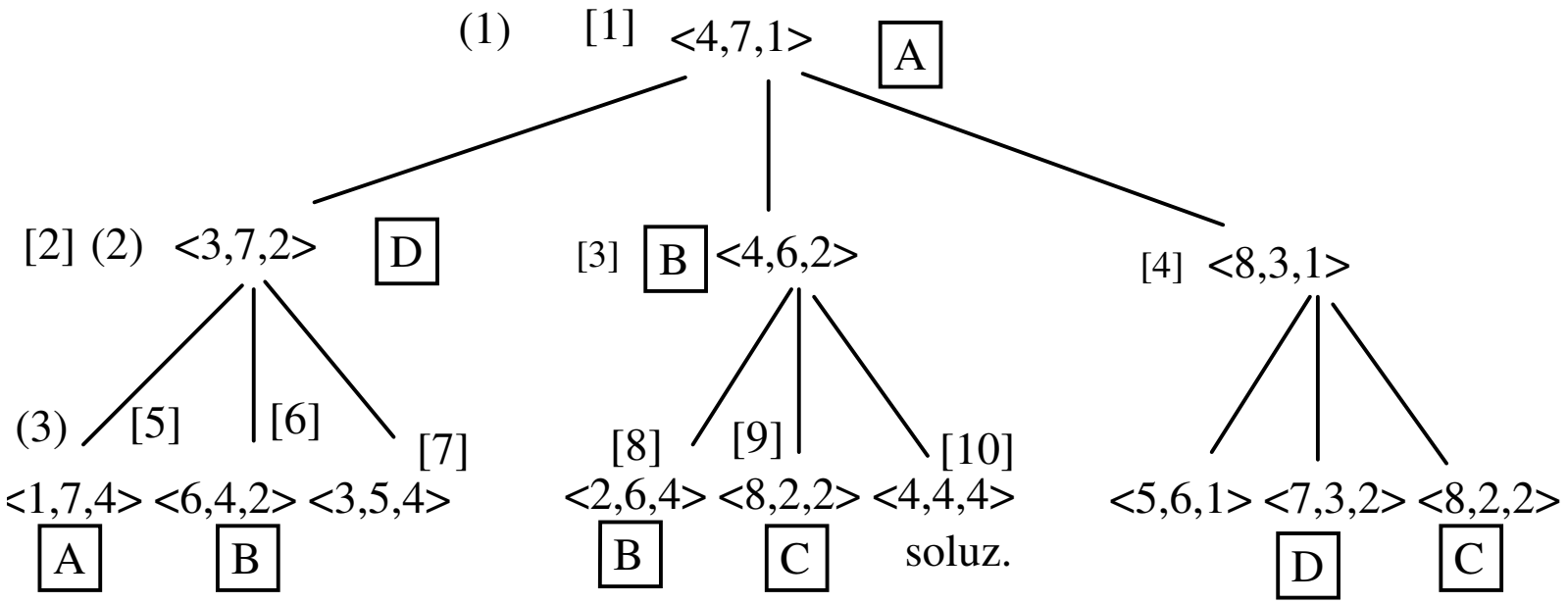


Esercizio – strategie di ricerca

- **Si supponga di avere 12 monete ripartite in tre distinte pile. La prima pila è formata da 4 monete, la seconda da 7 e la terza da 1. Lo spostamento delle monete deve avvenire secondo la seguente regola: ogni volta che si spostano delle monete da una pila A ad un'altra B, le monete in B devono raddoppiare di numero.**

- **Si mostri lo spazio di ricerca con stato iniziale rappresentato dalla tripla <4,7,1> e stato finale dalla tripla <4,4,4> generato in due spostamenti e lo si esplori con strategia in ampiezza e profondità.**

Soluzione – strategie di ricerca



Esercizio – strategie di ricerca

- **Si consideri un puzzle con la seguente configurazione iniziale:**

N	N	N	B	B	B	V

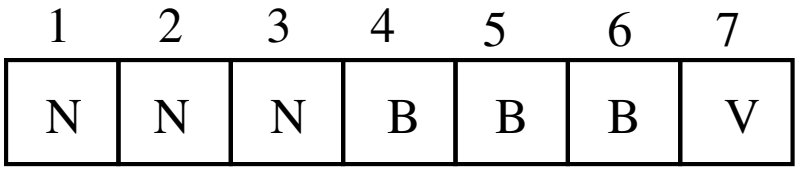
- **Ci sono tre tessere Nere (N) e tre tessere bianche (B) e una cella vuota (V).**
- **Il puzzle ha le mosse seguenti:**
- **(a) Una tessera si può spostare in una adiacente vuota con costo unitario.**
- **(b) Una cella si può spostare ed inserirsi nella cella vuota saltando al più due altre tessere con costo uguale alla distanza coperta (al numero delle tessere saltate più 1).**

Esercizio – strategie di ricerca

- **Il goal del puzzle è avere tutte le tessere bianche a sinistra di quelle nere, indipendentemente da dove sia la tessera vuota.**
- **Si costruisca una funzione euristica f = g + h' per questo problema dove g è stato specificato precedentemente e si mostri l'albero di ricerca prodotto dall'algoritmo A*.**

Esercizio – strategie di ricerca

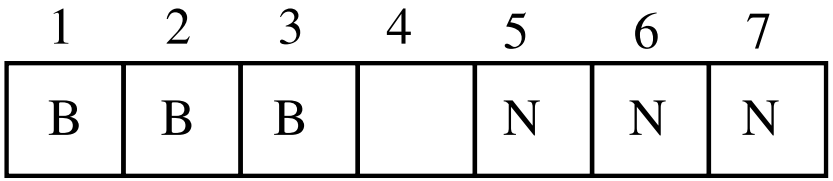
- **Esempi di EURISTICHE:**
- **1. Si può considerare il numero di tessere fuori posto: le tessere nere sono fuori posto se occupano le caselle 1, 2, 3 mentre le bianche sono fuori posto se occupano le celle 5, 6, 7.**



- **h'=5**

Esercizio – strategie di ricerca

- **Esempi di EURISTICHE:**
- **2. Come euristica si può considerare la distanza che separa le tessere nere dalle posizioni 4, 5, 6.**
- **Euristica non ammissibile: infatti, nella configurazione goal**

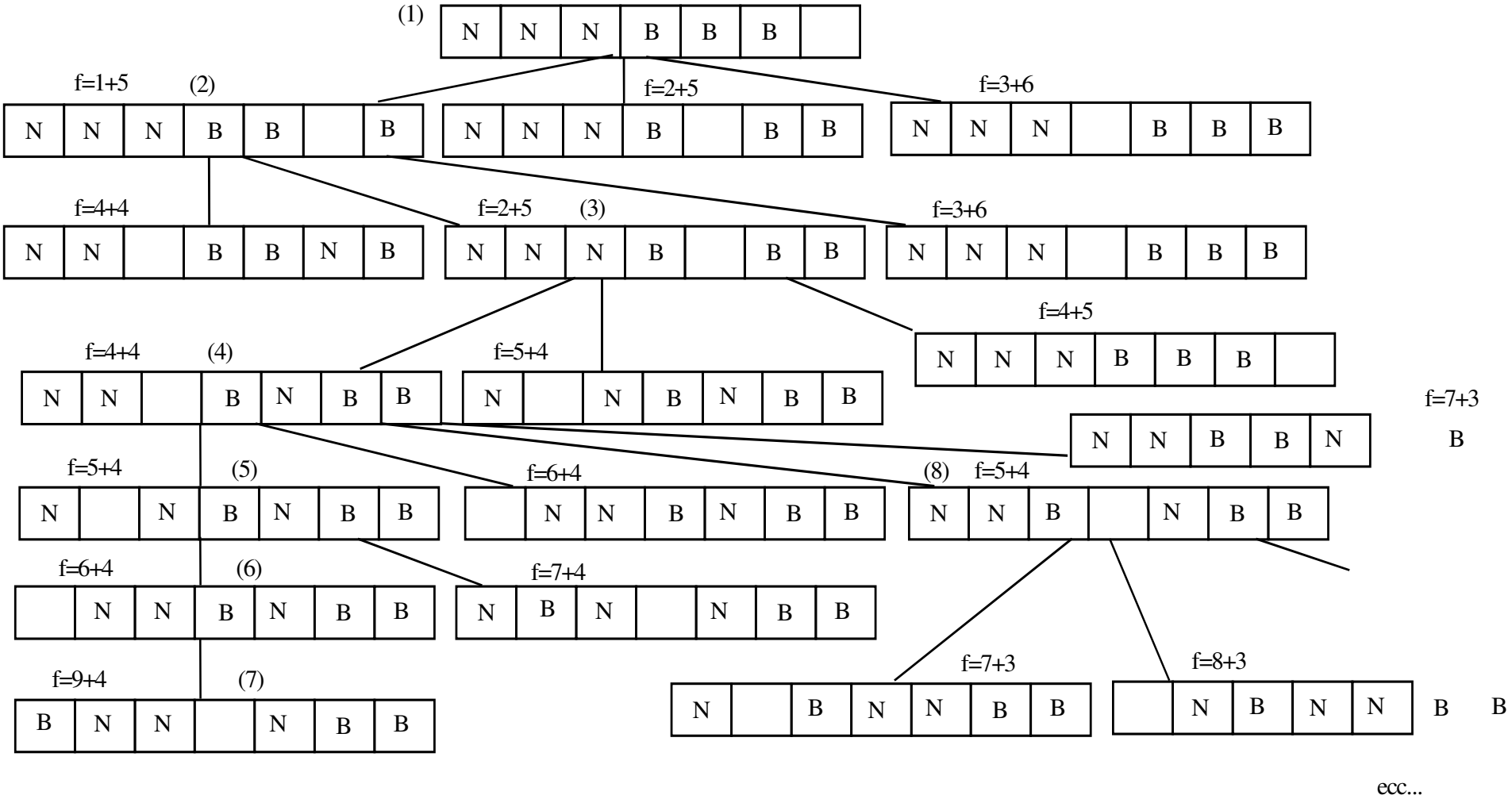


- **h=0**
- **h'=3**
- **il costo dell'euristica è 3 mentre il costo vero è 0 essendo già arrivati al goal.**

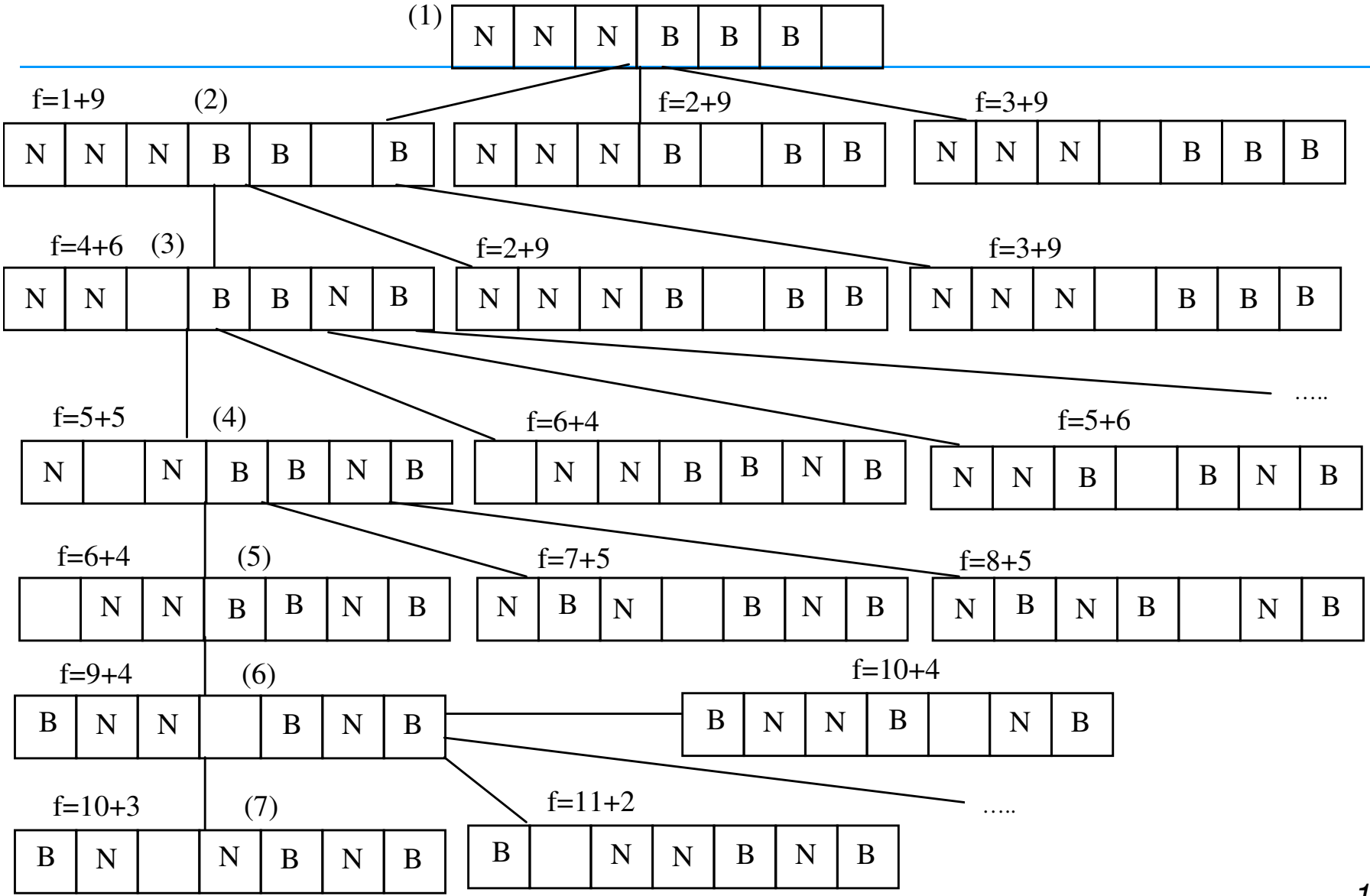
Esercizio – strategie di ricerca

- **Euristica ammissibile: min distanza delle tessere nere da qualunque posizione goal.**

Soluzione 1 – strategie di ricerca

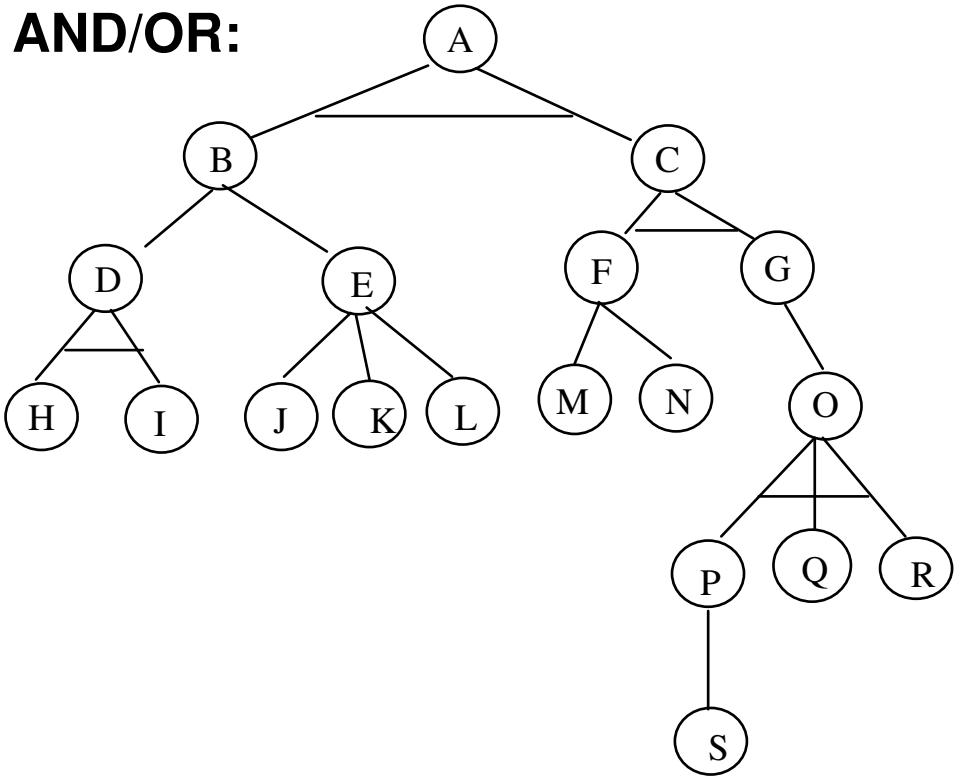


Soluzione 2 – strategie di ricerca



Esercizio – strategie di ricerca

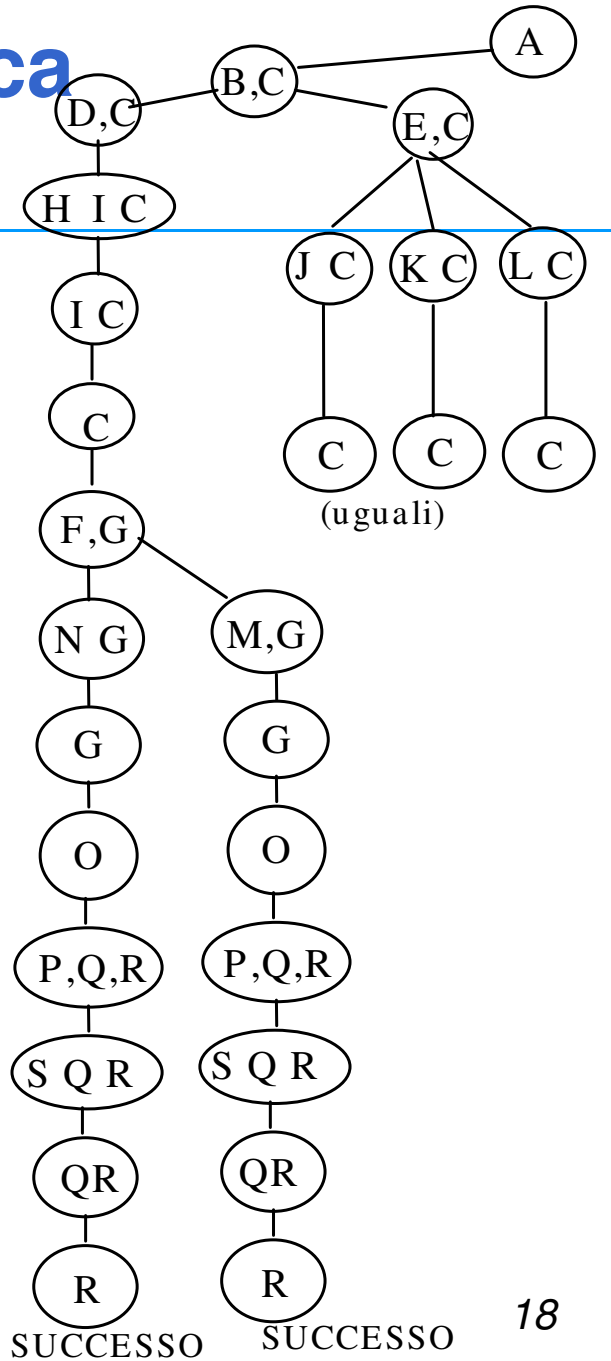
- **Dato il seguente albero AND/OR:**



- **Si indichino quali nodi devono essere di successo affinché venga dimostrato A. Si trasformi inoltre l'albero AND/OR in un albero OR equivalente, supponendo che tutte le foglie siano di successo. Esiste un solo albero OR equivalente o ne possono esistere più di uno?**

Soluzione – strategie di ricerca

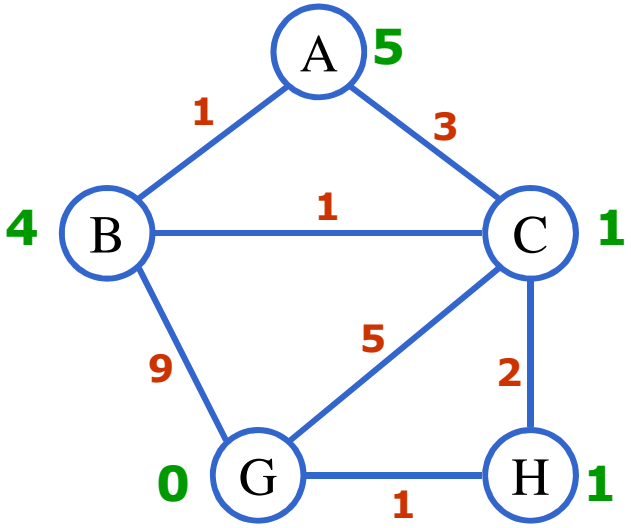
- **I nodi che devono essere di successo affinché si dimostri A sono H e I oppure J o K o L assieme a M o N con S Q ed R.**
- **Può esistere più di un albero OR, ma non cambiano i rami di successo.**



Esercizio – strategie di ricerca

- **Si consideri il seguente grafo, in cui i numeri sugli archi sono i costi e i numeri vicino agli stati sono le stime euristiche h. Si noti che gli archi sono non orientati e quindi intesi come percorribili da entrambe le parti. Si consideri lo stato A come stato iniziale e G come lo stato Goal.**

- **Si simuli la ricerca A* su questo grafo e si mostri l'albero generato supponendo di tenere traccia degli stati già espansi (ma non della strada ed il costo ad essi associato) per evitare loops.**

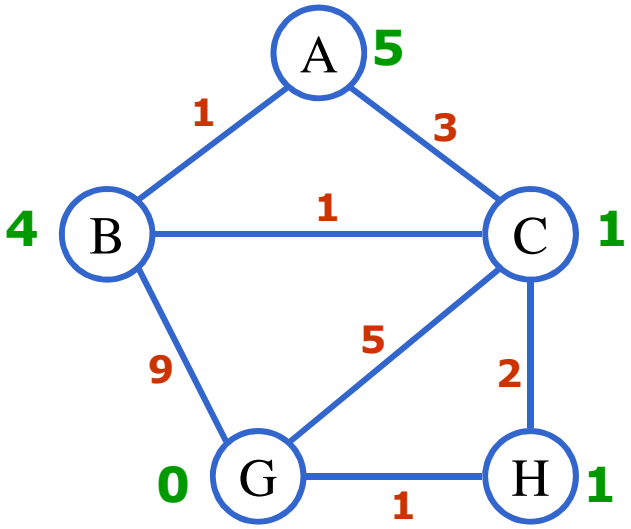


Esercizio – strategie di ricerca

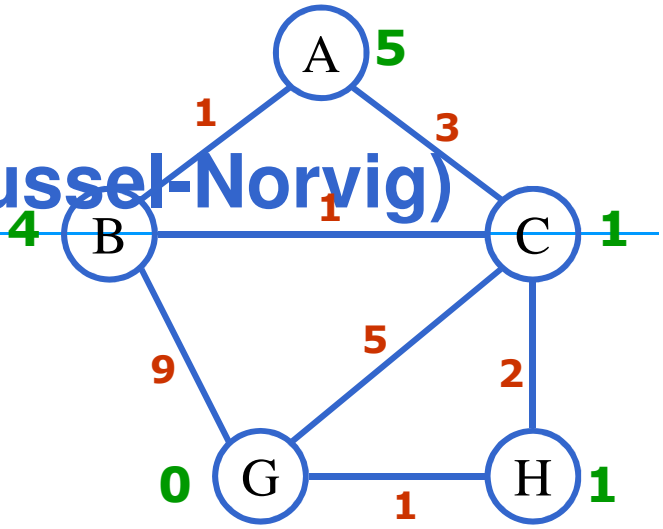
Si consideri il seguente grafo, in cui i numeri sugli archi sono i costi e i numeri vicino agli stati sono le stime euristiche h. Si noti che gli archi sono non orientati e quindi intesi come percorribili da entrambe le parti. Si consideri lo stato A come stato iniziale e G come lo stato Goal.

Si risponda alle seguenti domande:

1. **L'euristica presentata è ammissibile?**
2. **L'euristica presentata è consistente?**
3. **L'algoritmo trova la strada ottima? (a minor costo?) Per quale motivo? Se no, quale specifico cambiamento di valori euristici dello stato sarebbe sufficiente per renderla consistente?**



Esercizio – strategie di ricerca
Soluzione (usando A*grafi del Russel-Norvig)



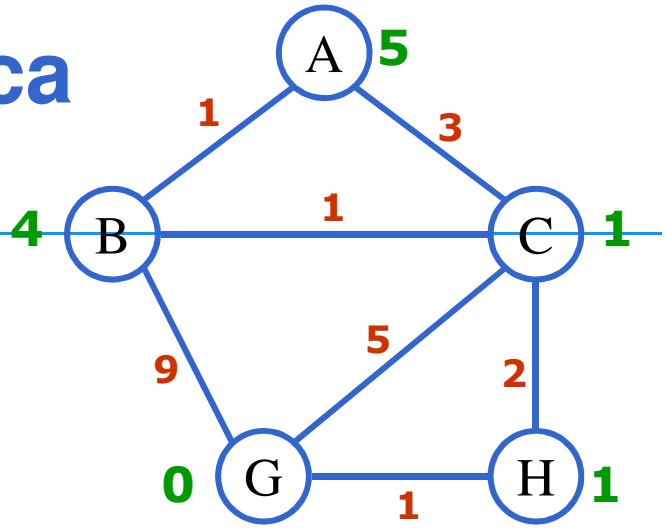
Percorso attuale	Costo	Stima	Lista Nodi Espansi
A	0	5	

0+5 = 5

Quale nodo scegliamo?

Trivial: Scegliamo il nodo A

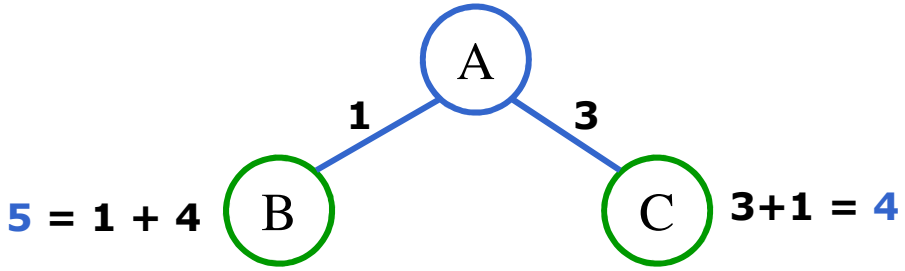
Esercizio – strategie di ricerca
Soluzione (usando A*grafi del Russel-Norvig)



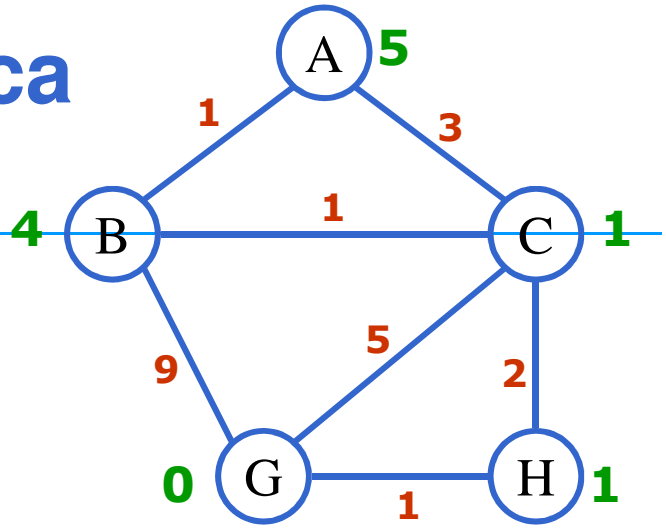
Percorso attuale	Costo	Stima	Lista Nodi Espansi
A	0	5	A

Quale nodo scegliamo?

→ Scegliamo il nodo C



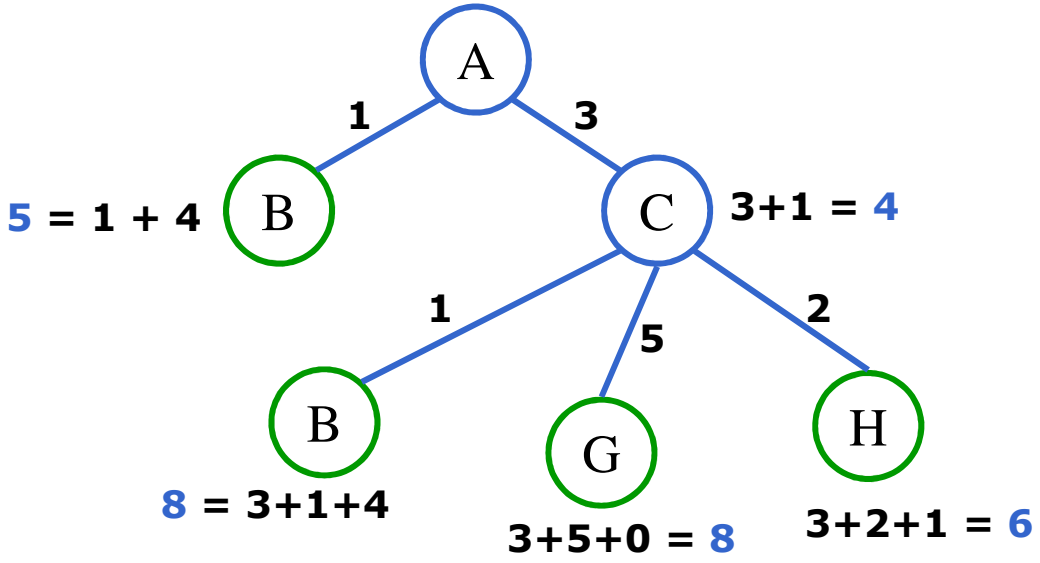
Esercizio – strategie di ricerca
Soluzione (usando A*grafi del Russel-Norvig)



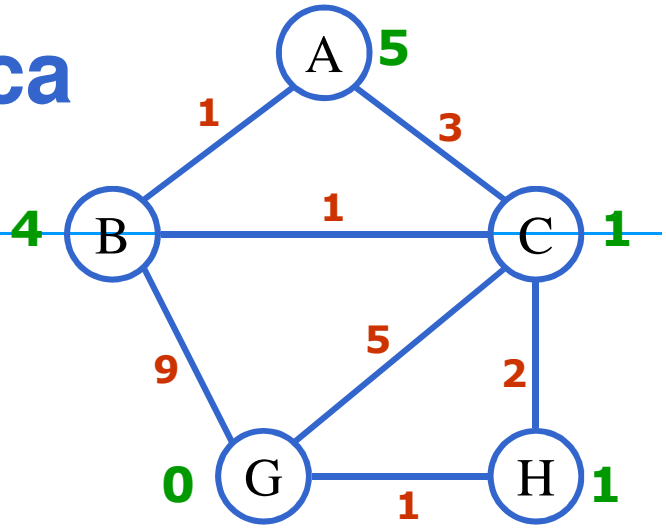
Percorso attuale	Costo	Stima	Lista Nodi Espansi
A	0	5	A
C A	3	4	C A

Quale nodo scegliamo?

→ **Scegliamo il nodo B**
(percorso B A)

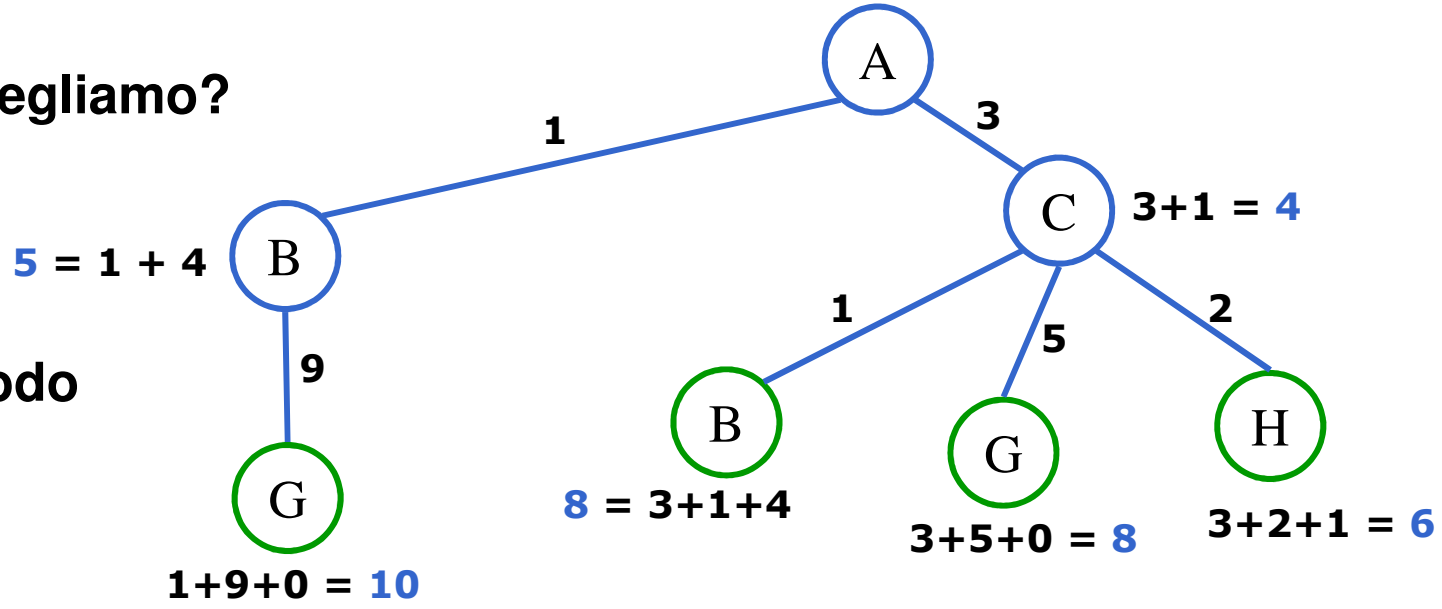


Esercizio – strategie di ricerca
Soluzione (usando A*grafi del Russel-Norvig)



Percorso attuale	Costo	Stima	Lista Nodi Espansi
A	0	5	A
C A	3	4	C A
B A	1	5	B C A

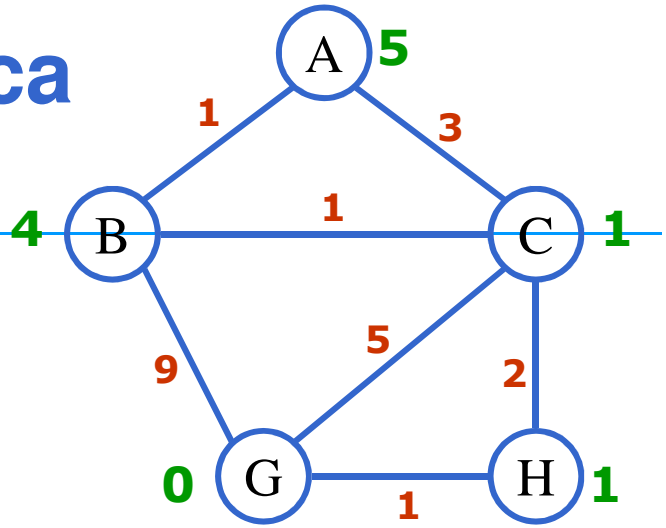
Quale nodo scegliamo?



→ **Scegliamo il nodo H**

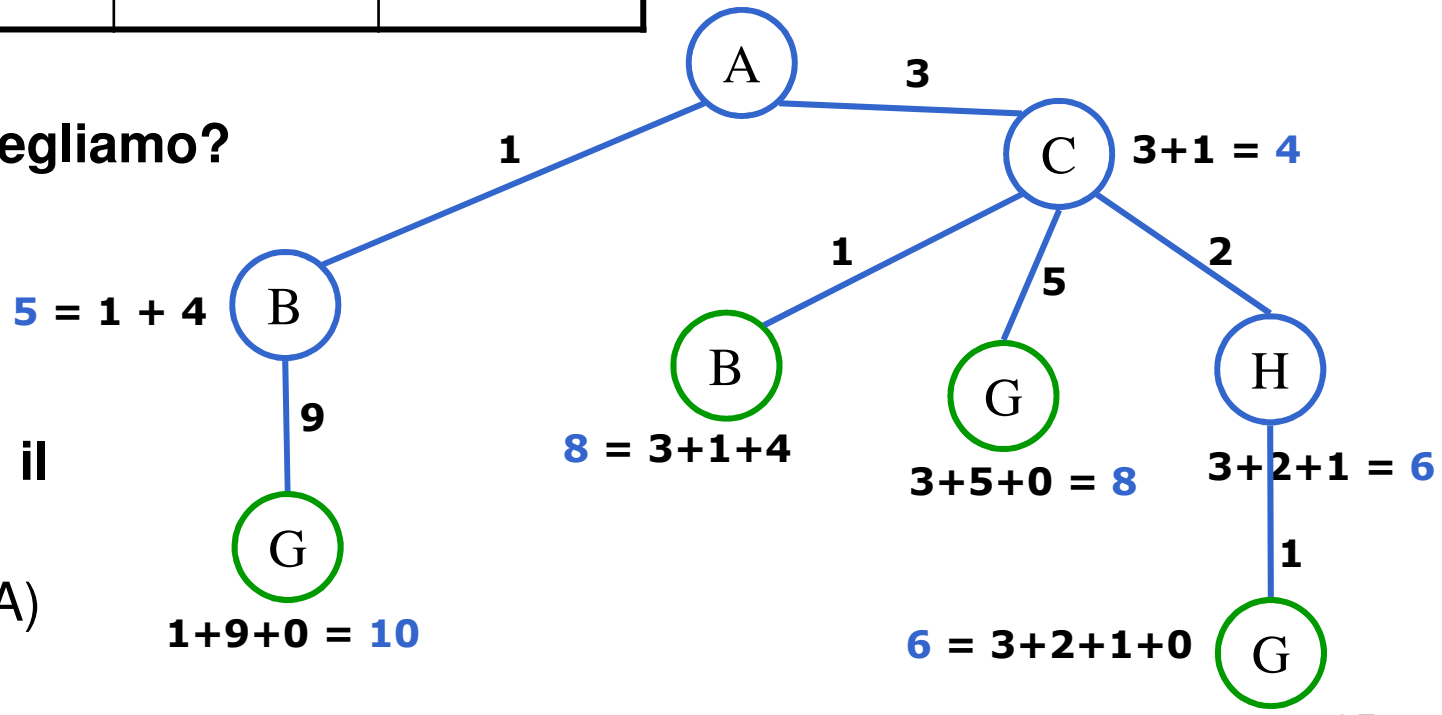
(percorso H C A)

Esercizio – strategie di ricerca
Soluzione (usando A*grafi del Russel-Norvig)



Percorso attuale	Costo	Stima	Lista Nodi Espansi
A	0	5	A
C A	3	4	C A
B A	1	5	B C A
H C A	5	6	H B C A

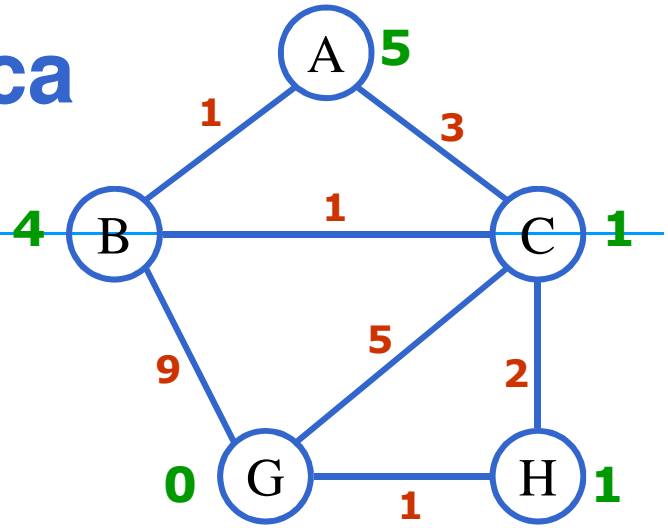
Quale nodo scegliamo?



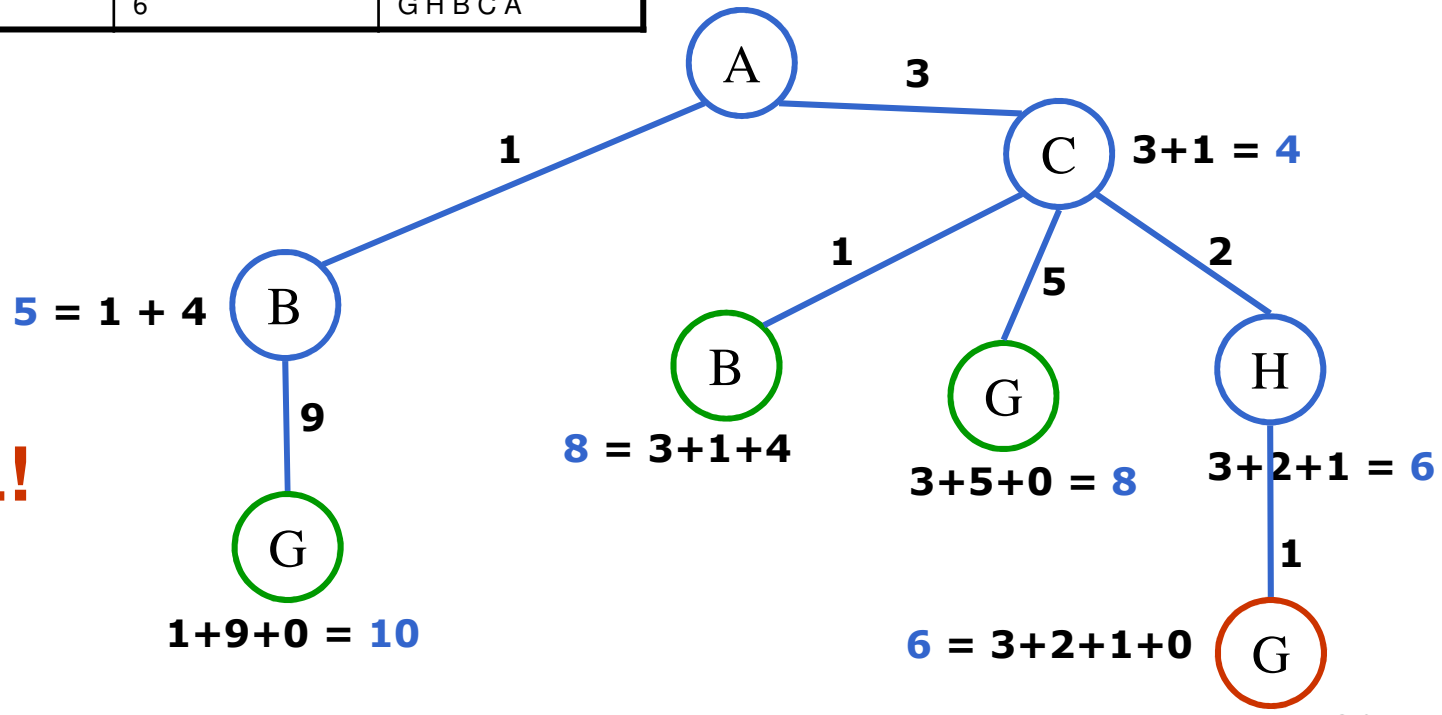
→ **Scegliamo il nodo G**

(percorso G H C A)

Esercizio – strategie di ricerca
Soluzione (usando A*grafi del Russel-Norvig)



Percorso attuale	Costo	Stima	Lista Nodi Espansi
A	0	5	A
C A	3	4	C A
B A	1	5	B C A
H C A	5	6	H B C A
G H C A	6	6	G H B C A



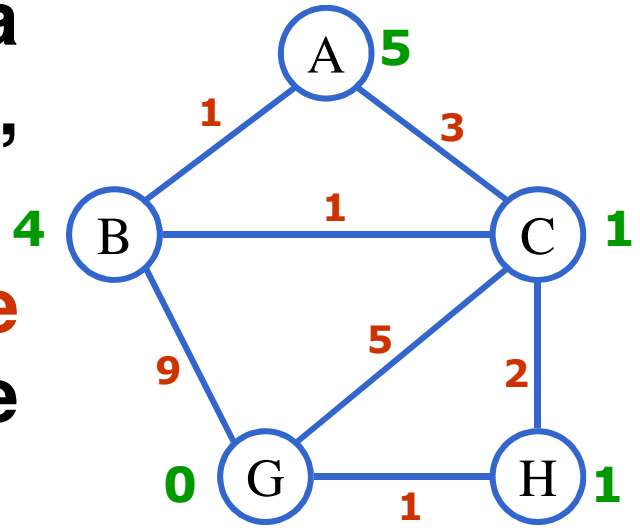
G è il GOAL!

Esercizio – strategie di ricerca
Alcune osservazioni…

La soluzione selezionata dall'algoritmo è stata: ACHG, con costo 6.

Tale soluzione non è la soluzione ottima. Infatti la soluzione ABCHG ha costo 5

Ciò è dovuto al fatto che l'euristica non è consistente. In tal caso, l'algoritmo A* per grafi non garantisce di trovare la soluzione ottima.



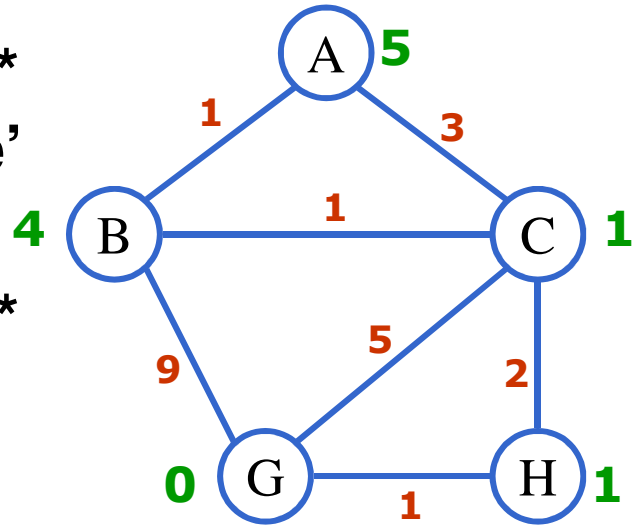
Esercizio – strategie di ricerca
Alcune osservazioni…

L'ottimalità nel caso dell'algoritmo A* per grafi è garantita se l'euristica e' consistente.

Si può ottenere quindi l'ottimalità di A* per grafi in due modi:

1. **Scegliendo una euristica consistente**

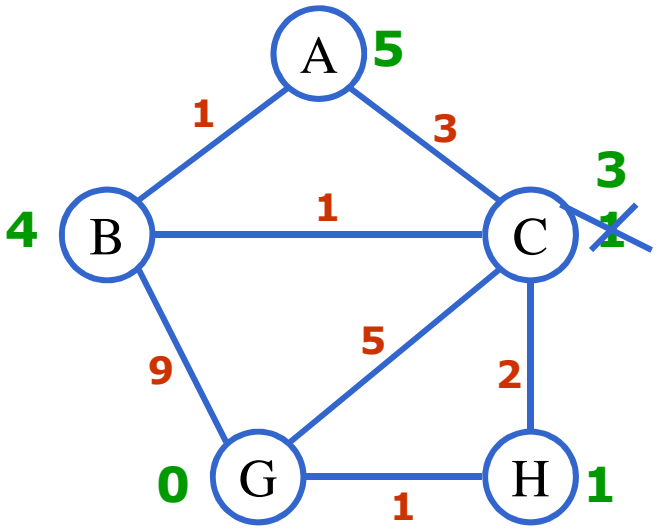
2. **Modificando l'algoritmo, tramite l'aggiornamento del costo per giungere in un nodo quando si trovano percorsi migliori. Tutti gli eventuali costi di altri percorsi devono poi essere aggiornati in maniera opportuna… vedi algoritmo modificato nelle slides del corso!!!!**



Esercizio – strategie di ricerca

- **L'euristica in questo esempio è inconsistente, in particolare se consideriamo i costi tra A e C, e poi anche i costi tra B e C.**
- **Supponendo che il valore di euristica di C sia ad esempio 3, allora l'euristica risulta essere consistente, e quindi l'A* per grafi selezionerà il cammino ottimo.**

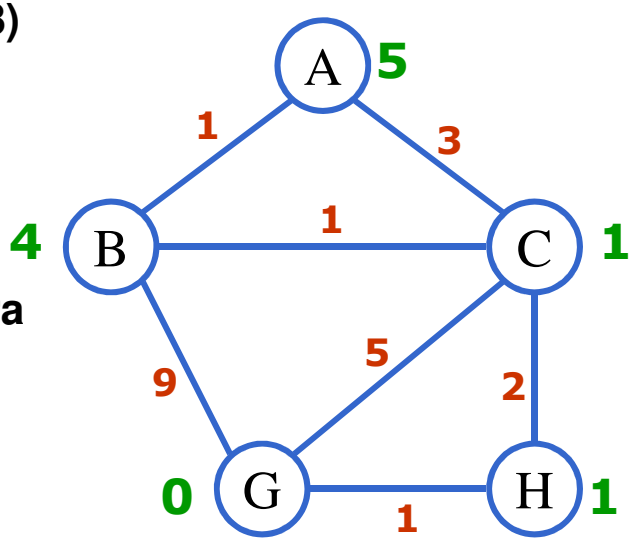
1. **La=[A], Lc=[ ]**
2. **La = [(AB,5), (AC,6)]**
   Lc = [A]
3. **La = [(ABC,5), (AC,6), (ABG,10)]**
   Lc = [A,B]
4. **La = [(ABCH,5), (AC,6), (ABCG,7), (ABG,10)]**
   Lc = [A,B,C]
5. **La = [(ABCHG,5), (AC,6), (ABCG,7), (ABG,10)]**
   Lc = [A,B,C,H]
6. **Seleziono G e mi accorgo che è il goal**



Esercizio – strategie di ricerca

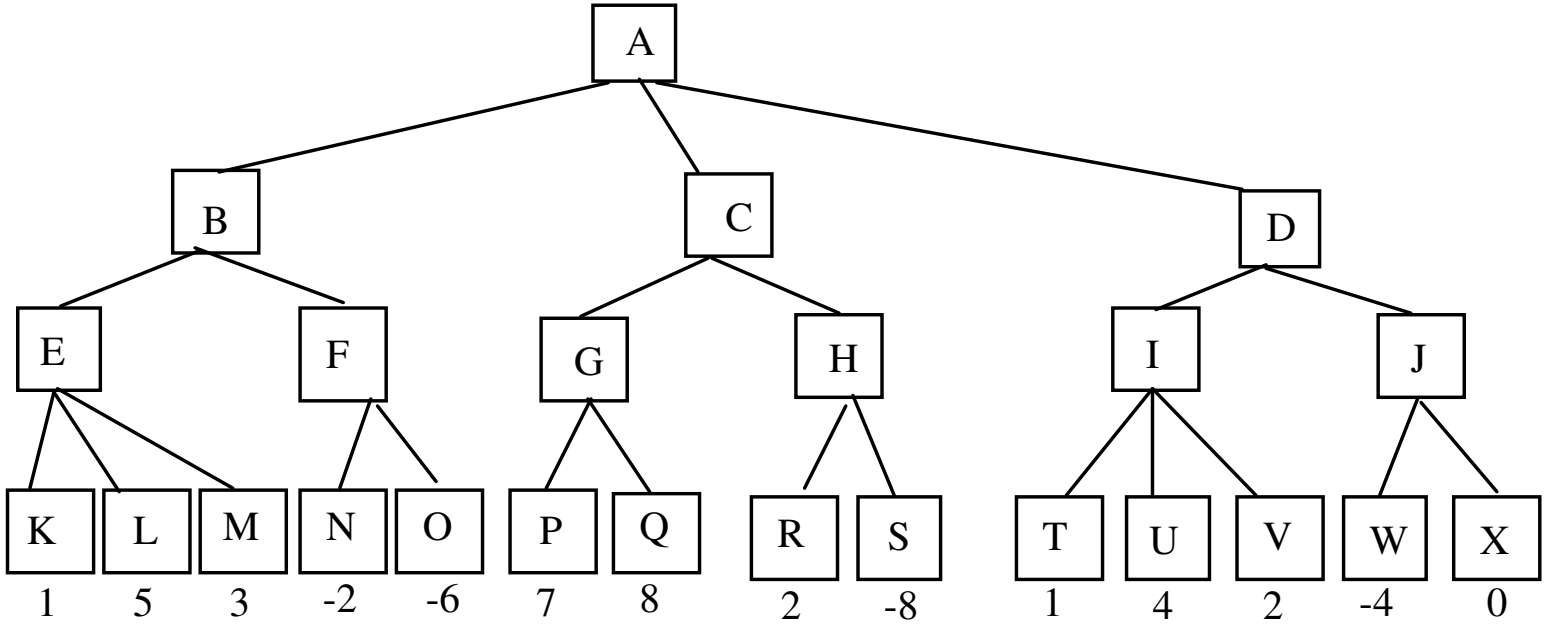
- **Volendo invece usare l'algoritmo modificato di A*:**

1. **La=[(A,5)], Lc=[ ]**
2. **La = [(AC,4), (AB,5)]**
   Lc = [(A,5)]
3. **La = [(AB, 5) ,(ACH,6), (ACG,8)] Nota: il nodo (ACB,8) non e' inserito perche' (AB, 5)**
   Lc = [(A,5),(C,4)]
4. **La = [(ABCH,5), (ABCG,7)]**
   Nota: ho trovato un percorso migliore (ABC,3) per giungere a C, e quindi aggiorno tutti i nodi in maniera opportuna, col nuovo percorso; siccome C e' tra i nodi chiusi, non metto (ABC,3) in La; (ABG,10) peggiora l'attuale valore per G e quindi non lo aggiungo;
   Lc = [(A,5),(C,3), (B,5)]
5. **La = [(ABCHG,5)] Trovato un nuovo percorso migliore per G**
   Lc = [(A,5),(C,3),(B,5),(H,5)]
6. **Seleziono G e mi accorgo che è il goal**



Esercizio – min-max e tagli alfa-beta

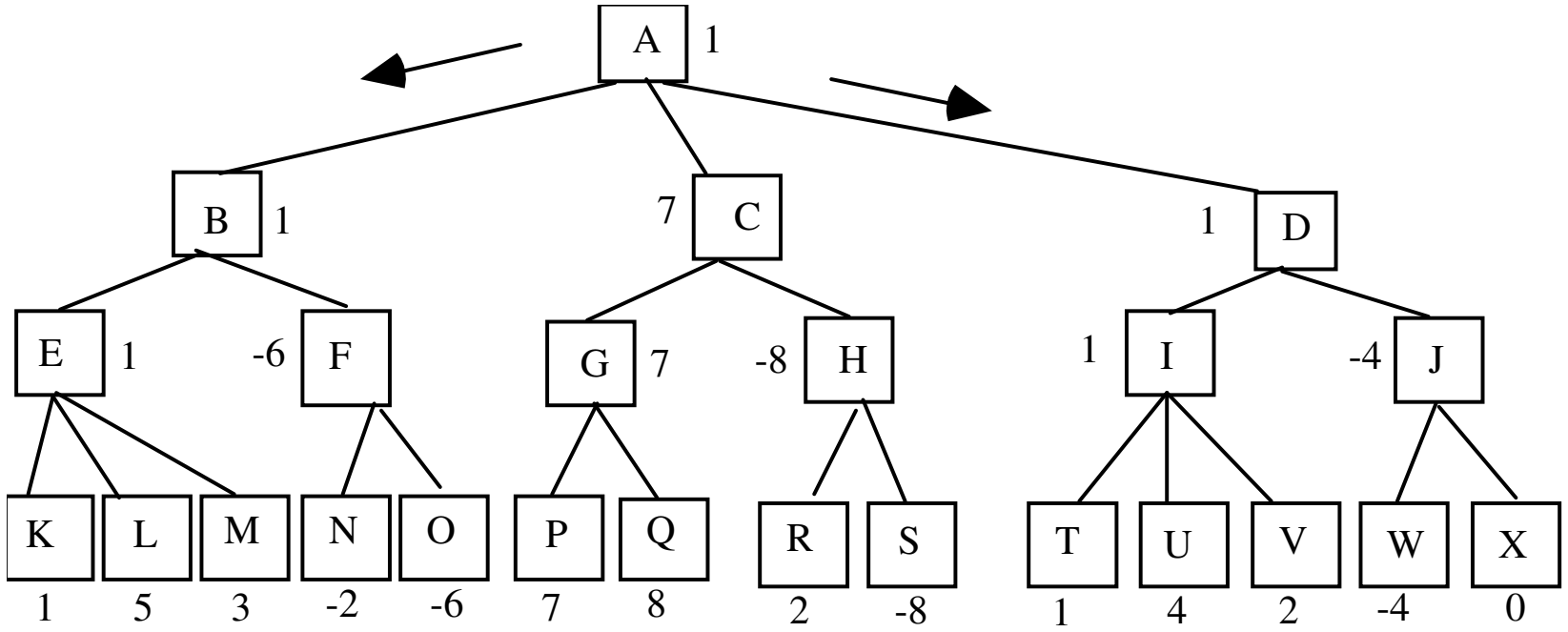
- Si consideri il seguente albero di gioco dove i punteggi sono tutti dal punto di vista del primo giocatore.



- Supponendo che il primo giocatore sia MIN, quale mossa dovrebbe scegliere? Si applichi l'algoritmo MIN-MAX. Successivamente si mostrino i tagli alfa-beta.

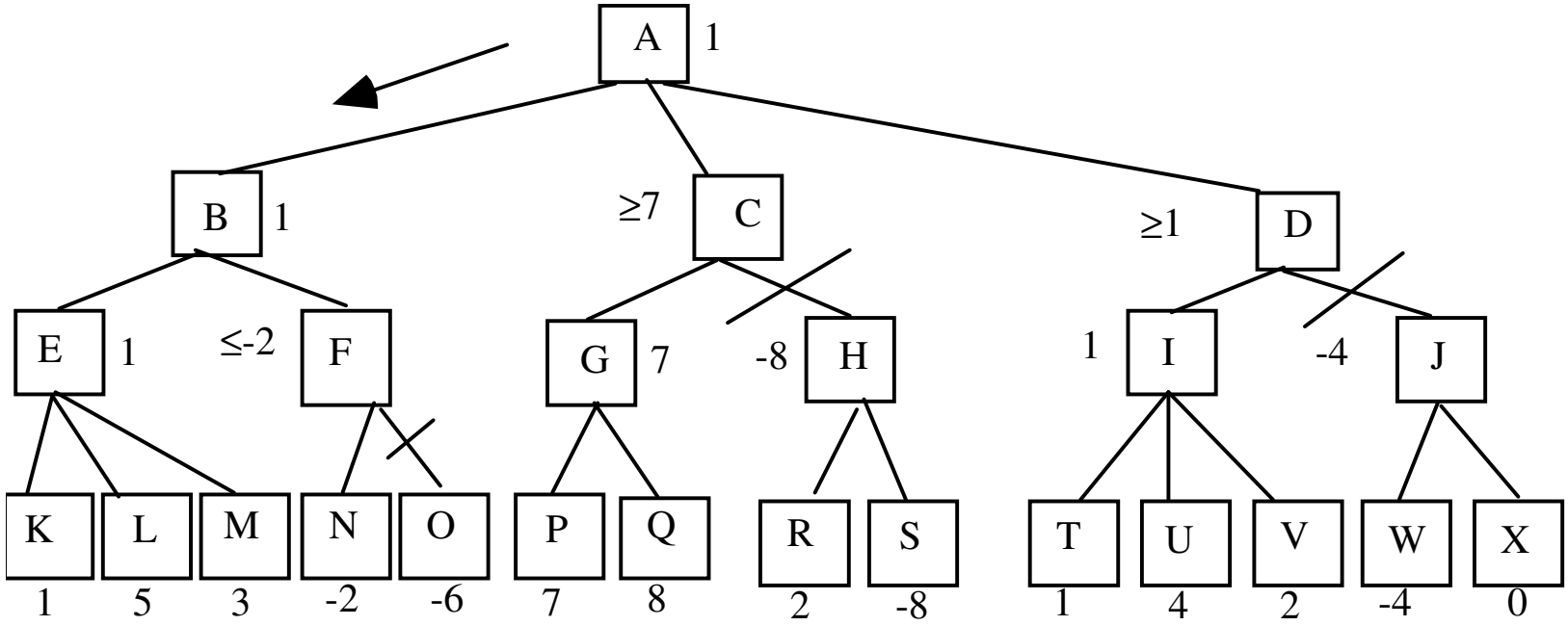
Soluzione – min-max e tagli alfa-beta

- Min-Max:



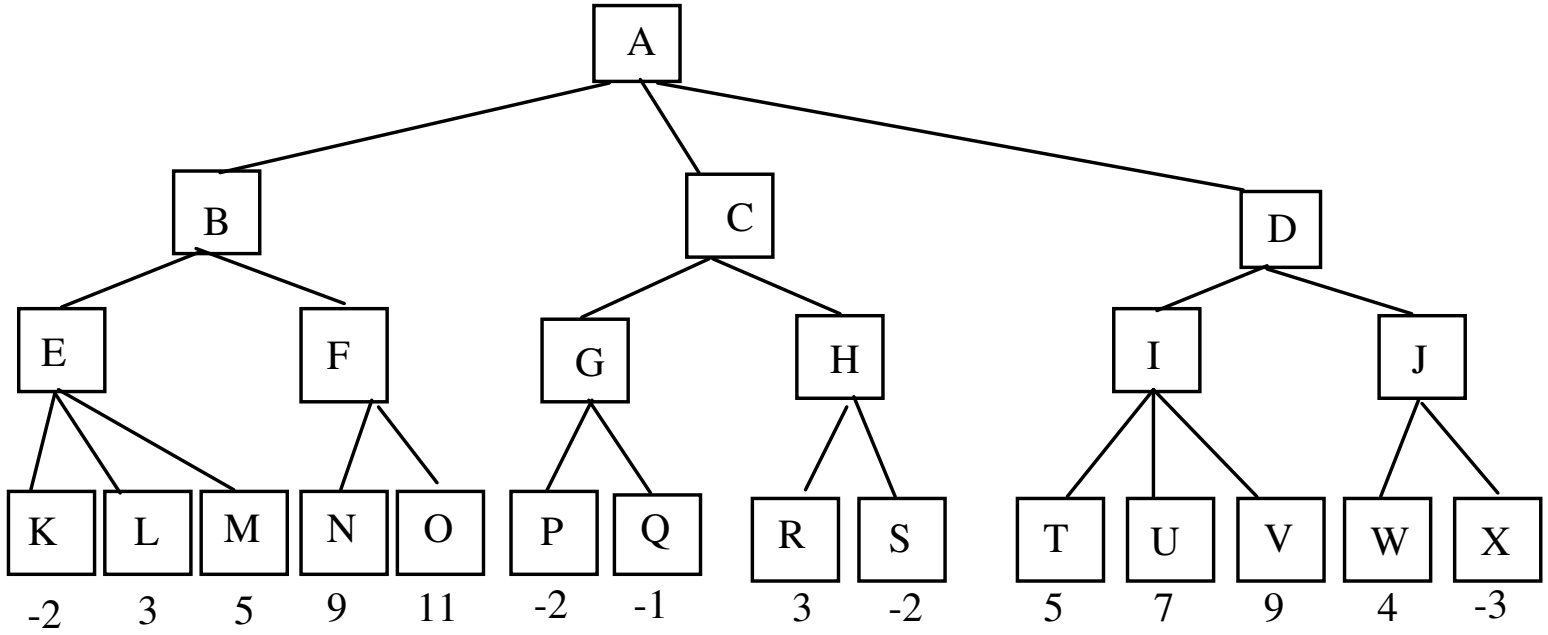
Soluzione – min-max e tagli alfa-beta

- Tagli alfa-beta:



Esercizio – min-max e tagli alfa-beta

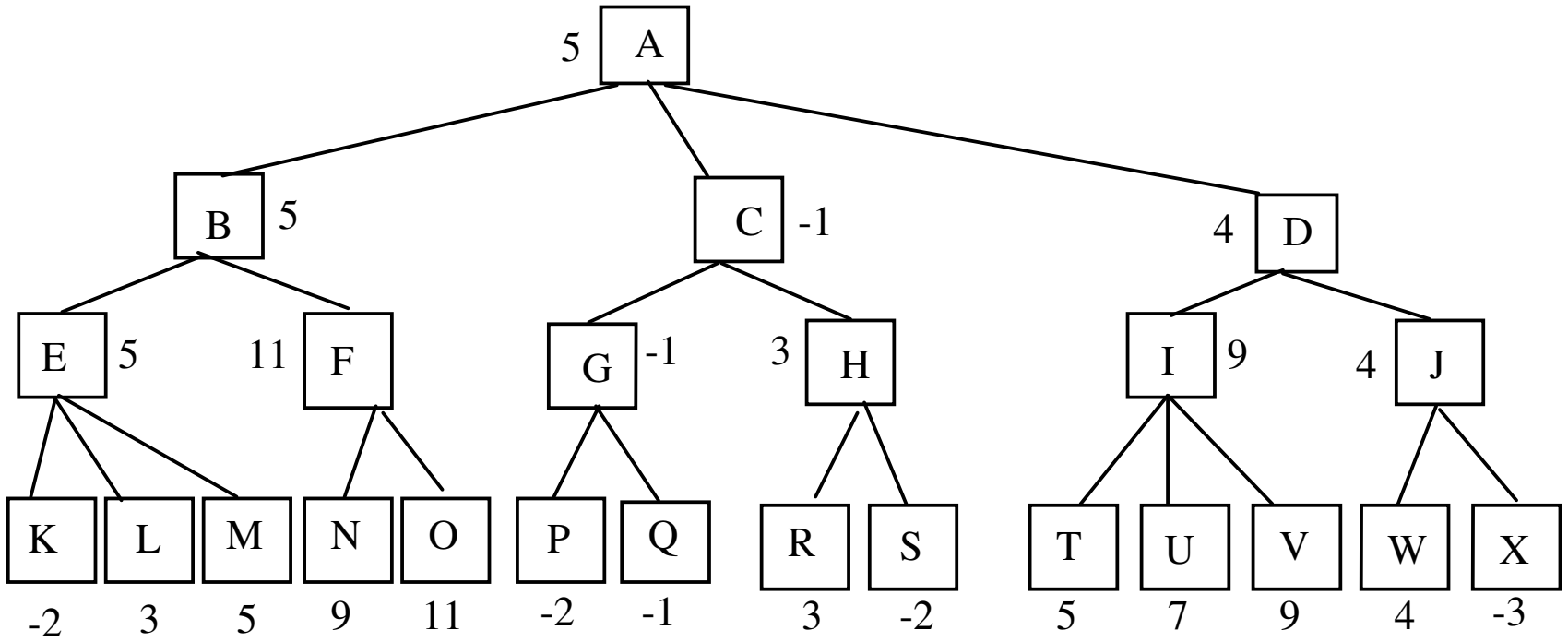
- Si consideri il seguente albero di gioco dove i punteggi sono tutti dal punto di vista del primo giocatore.



- Supponendo che il primo giocatore sia MAX, quale mossa dovrebbe scegliere? Si applichi l'algoritmo MIN-MAX. Successivamente si mostrino i tagli alfa-beta.

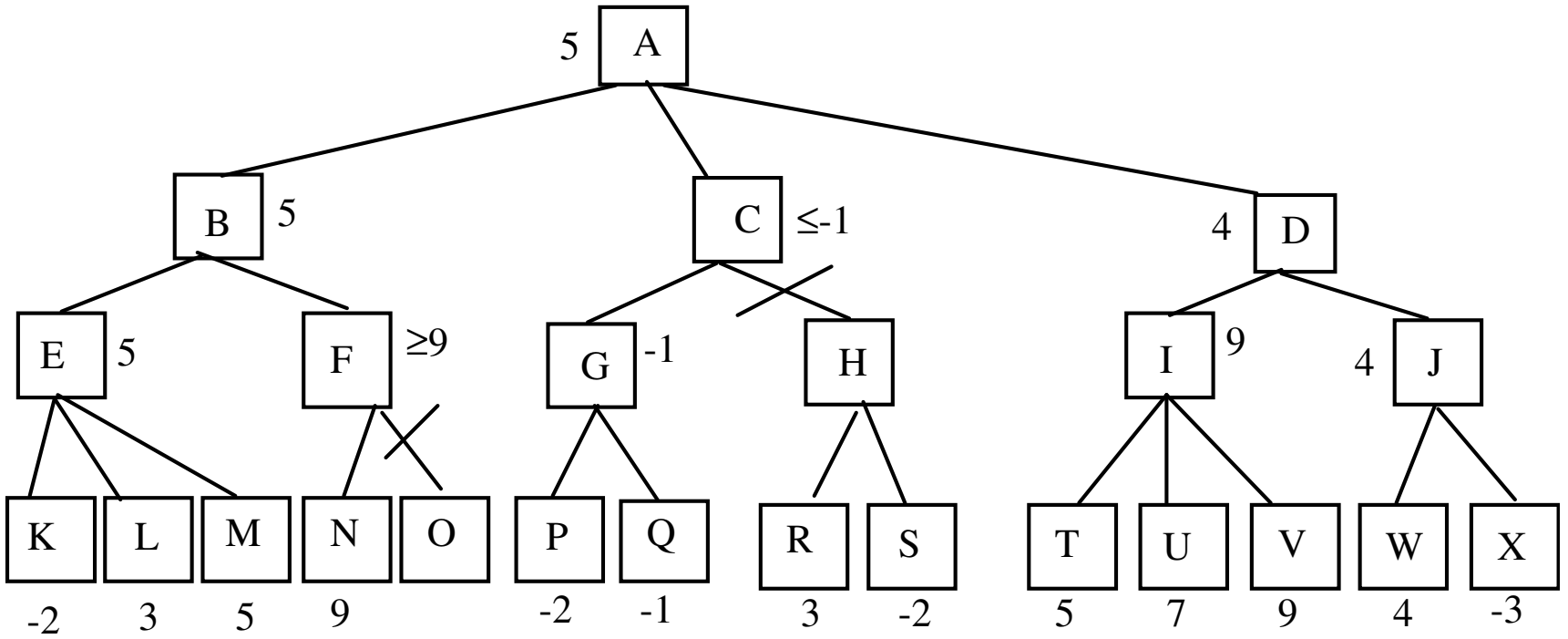
Soluzione – min-max e tagli alfa-beta

- Min-Max:

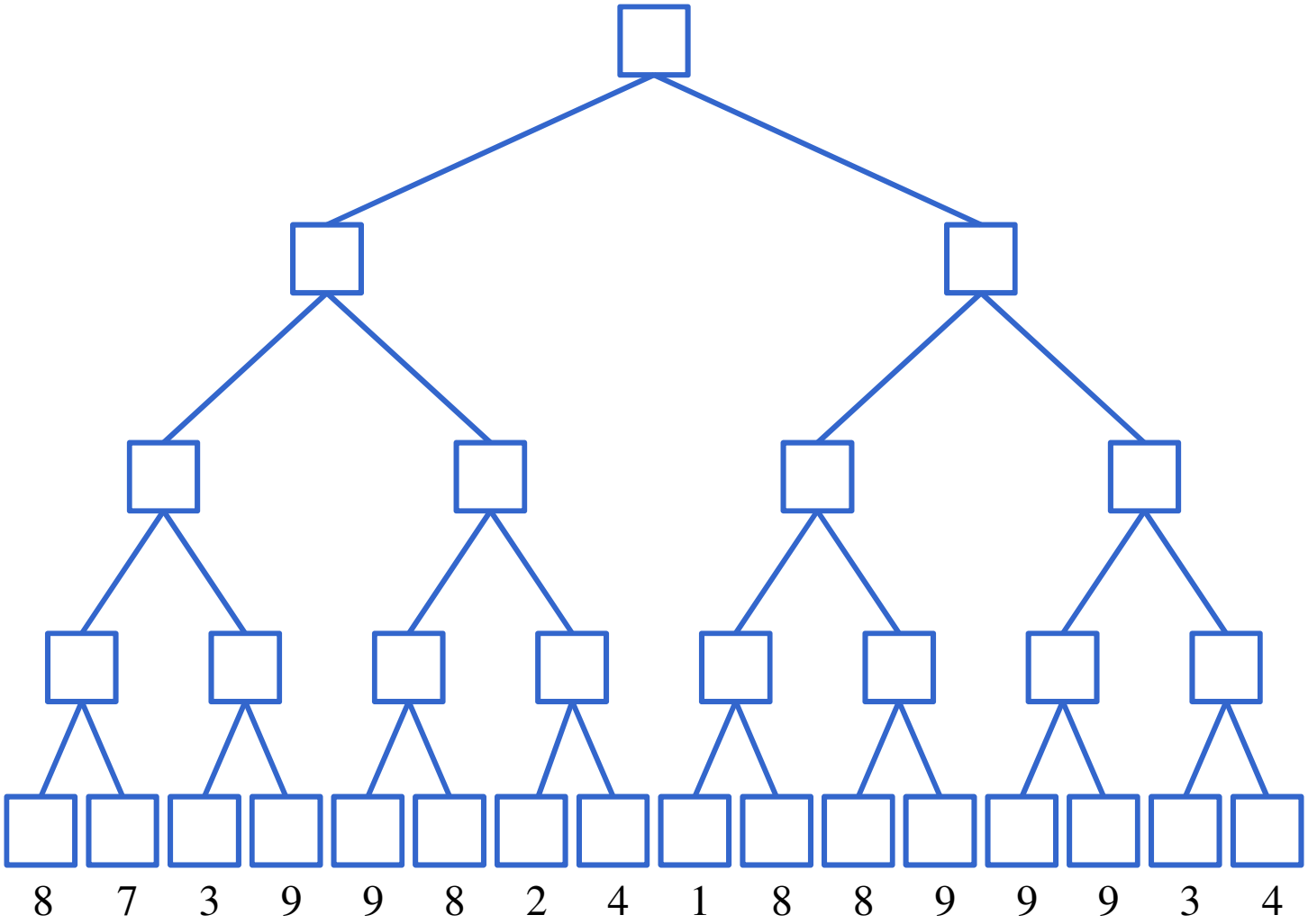


Soluzione – min-max e tagli alfa-beta

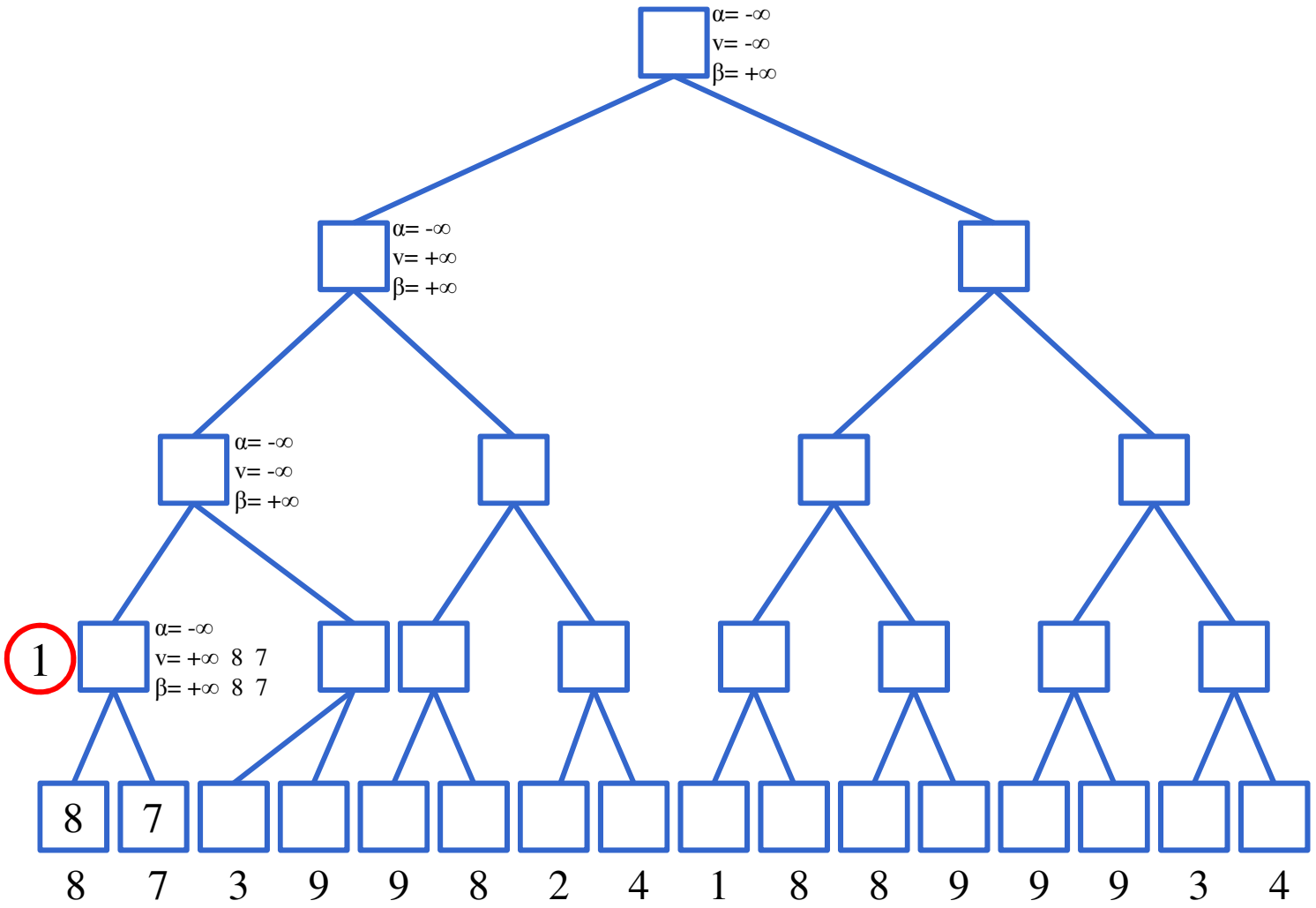
- Tagli alfa-beta:



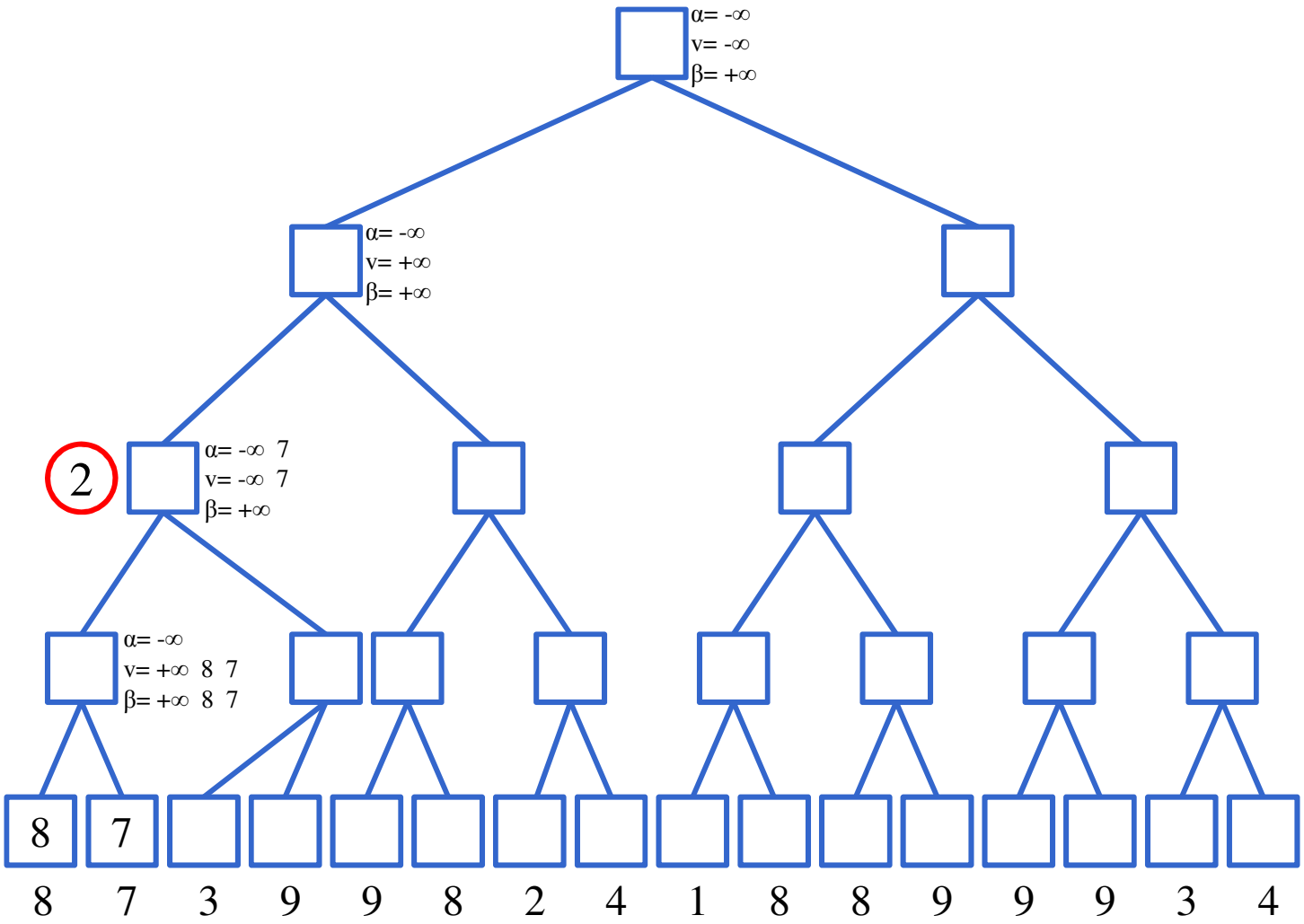
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



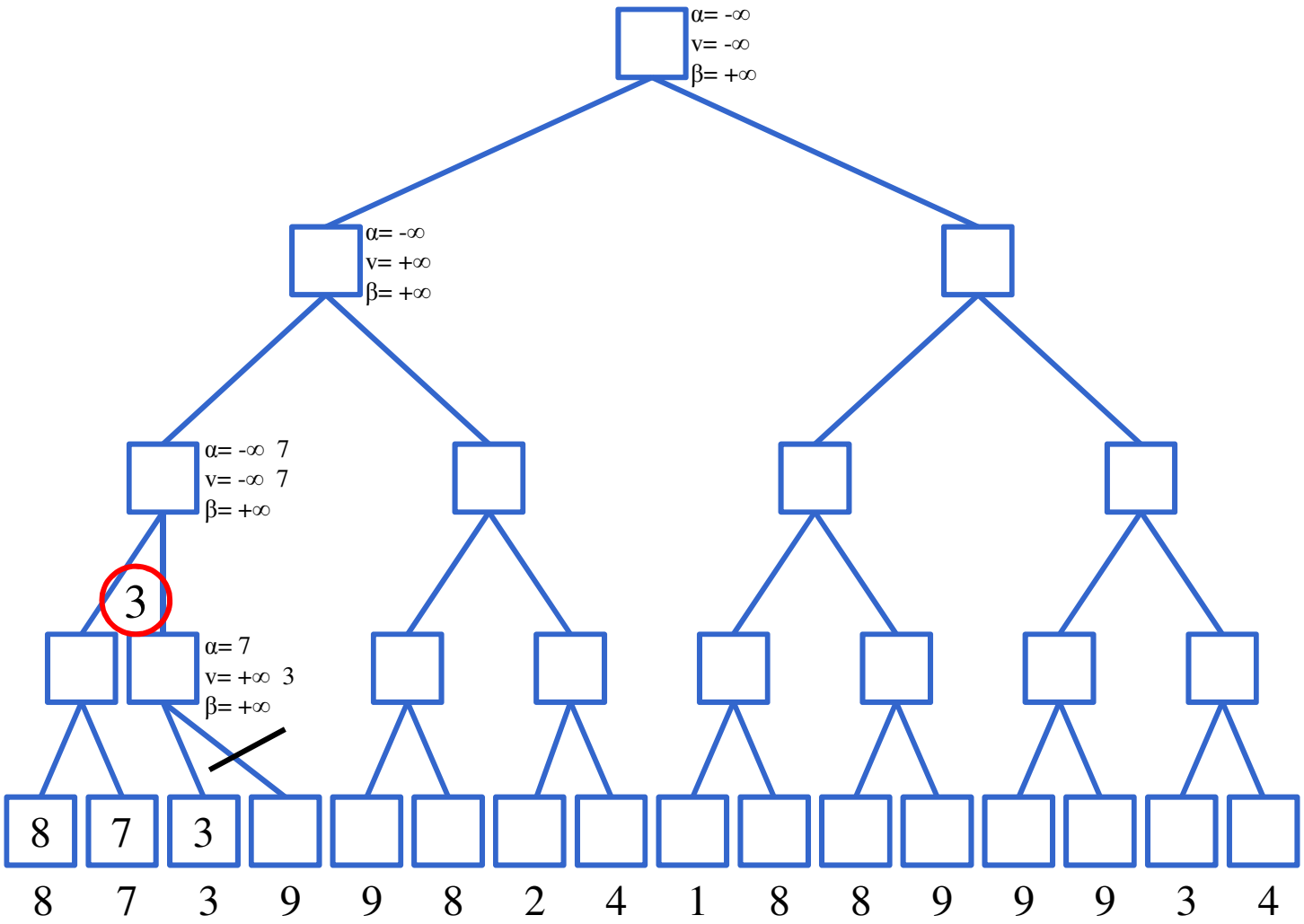
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



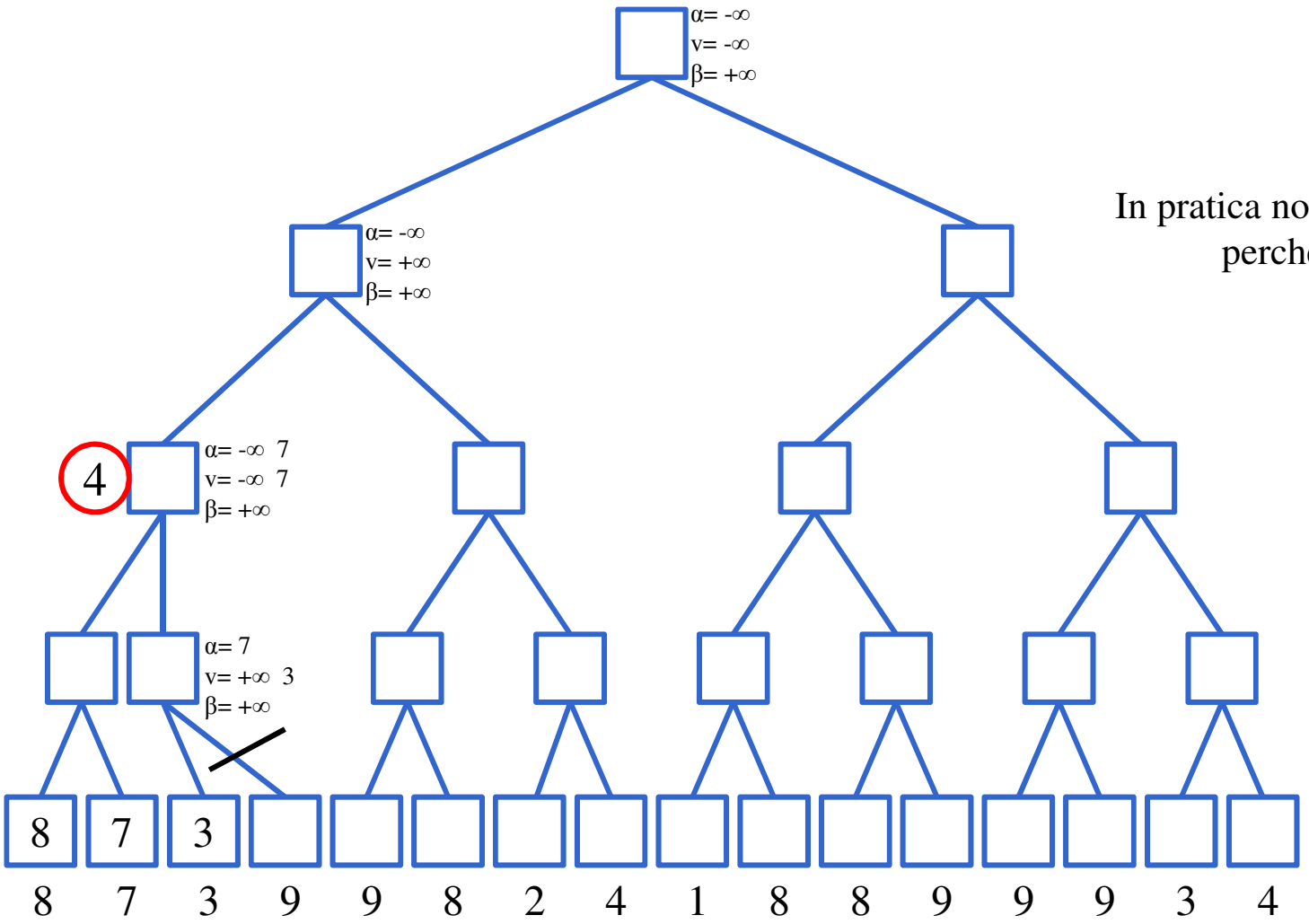
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011

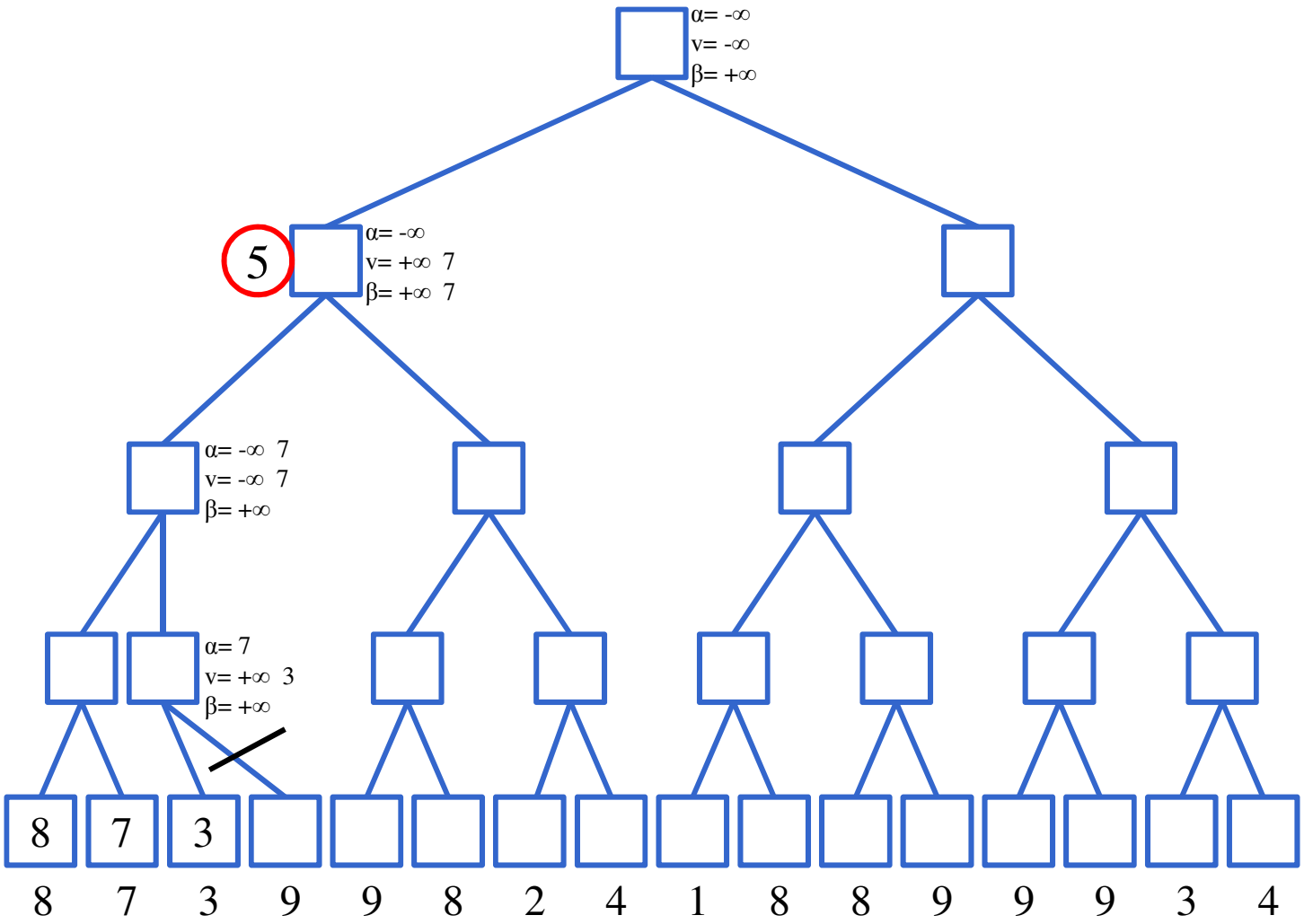


Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011

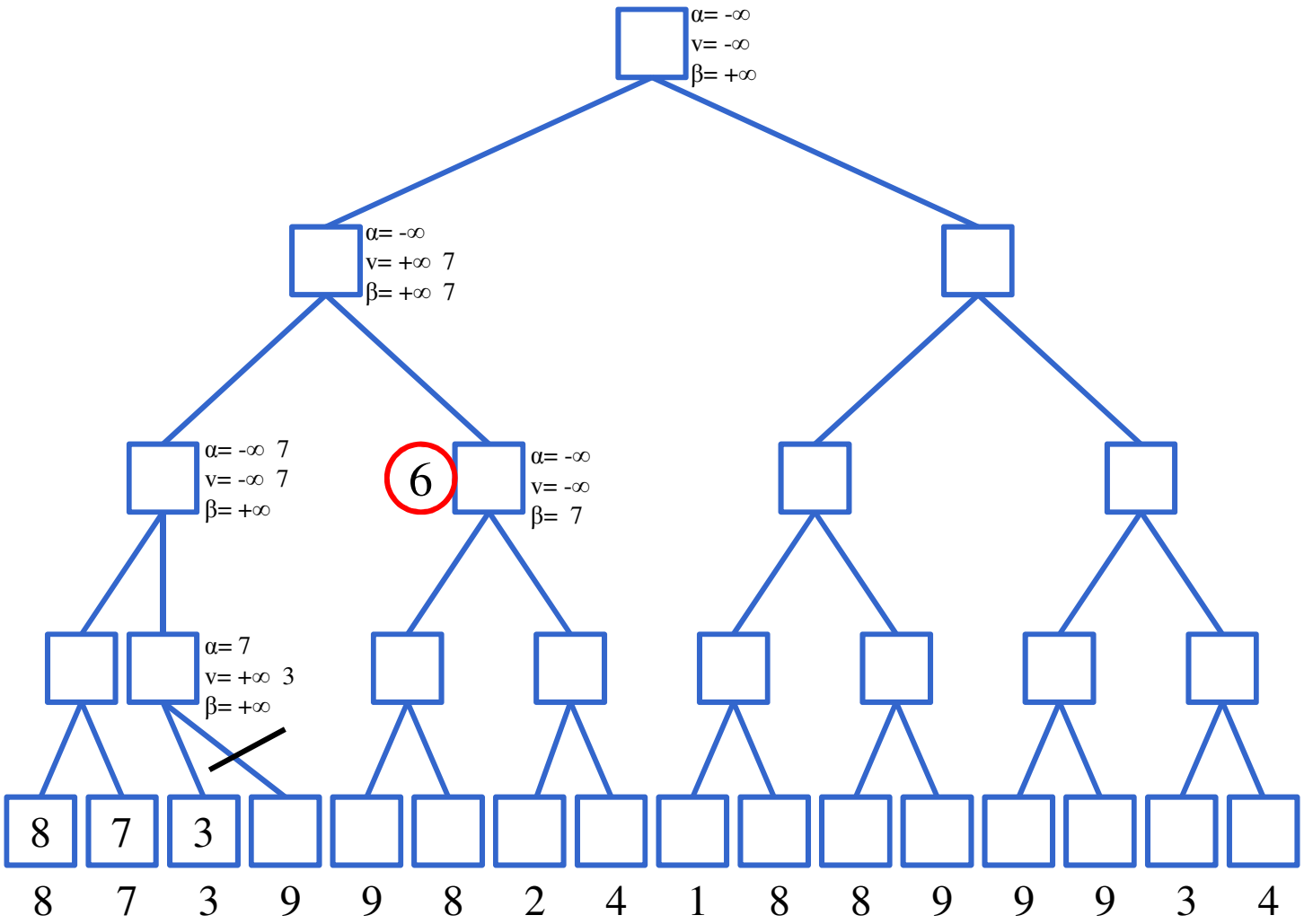


In pratica non accade nulla, perché 3<7…

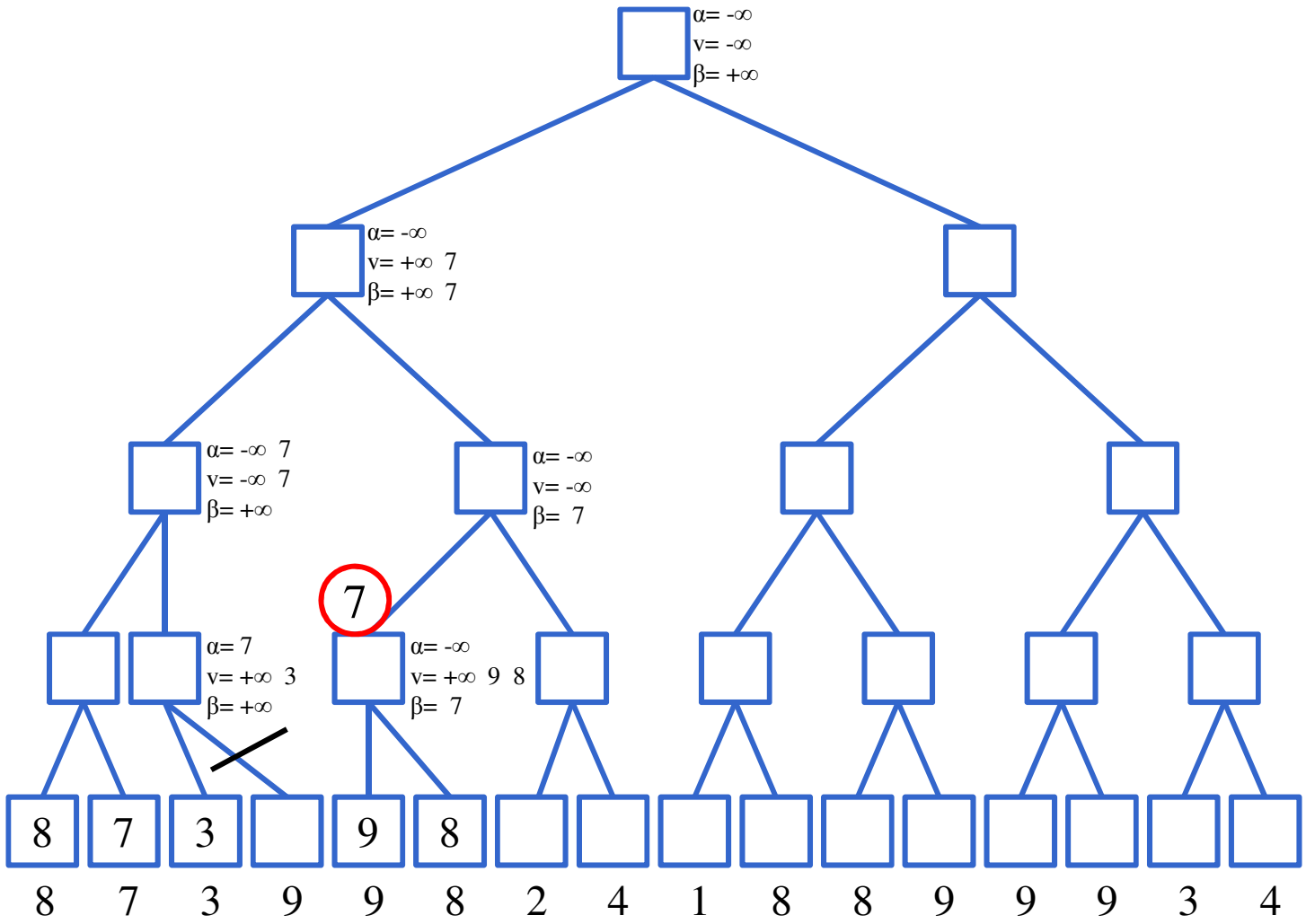
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



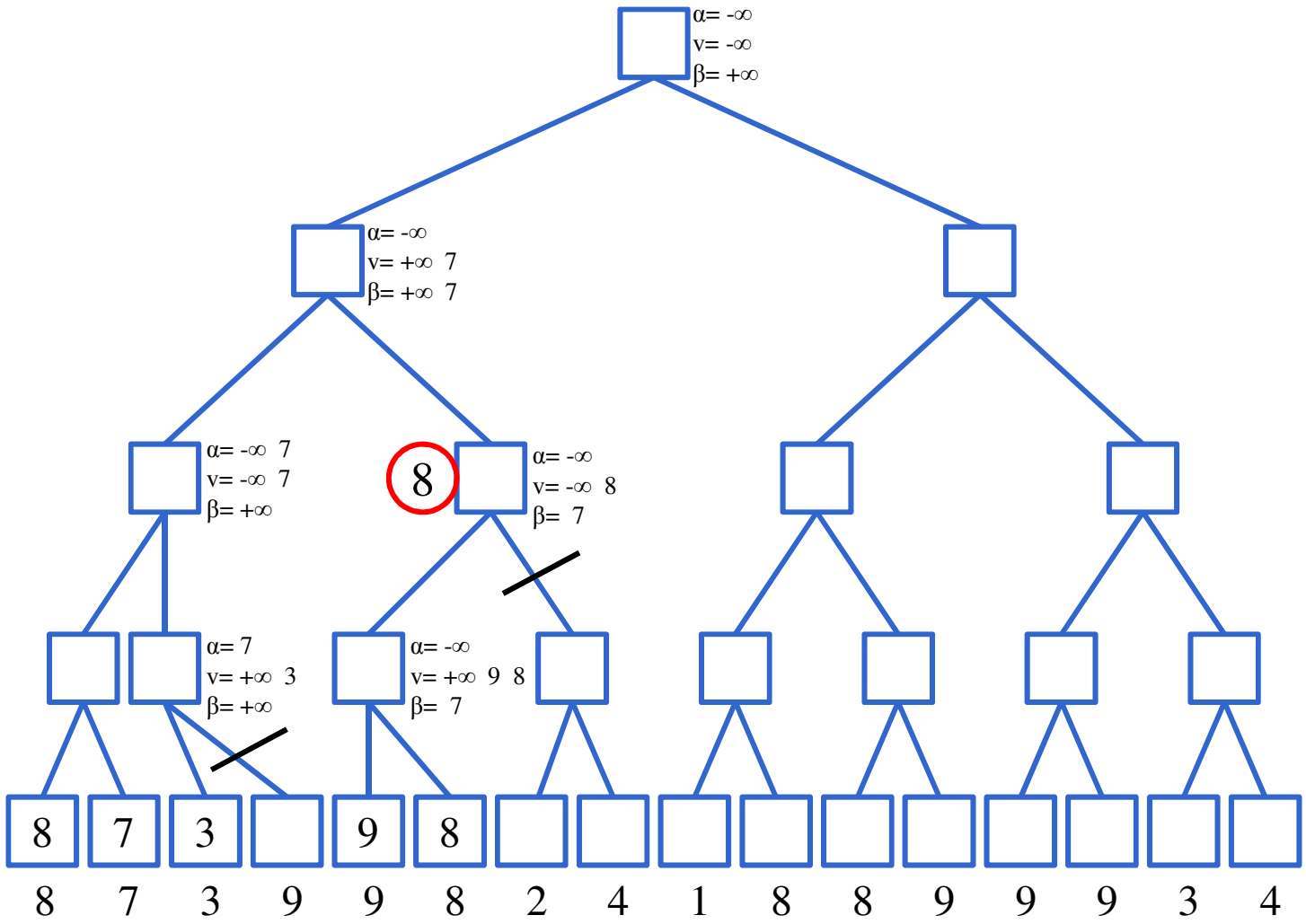
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



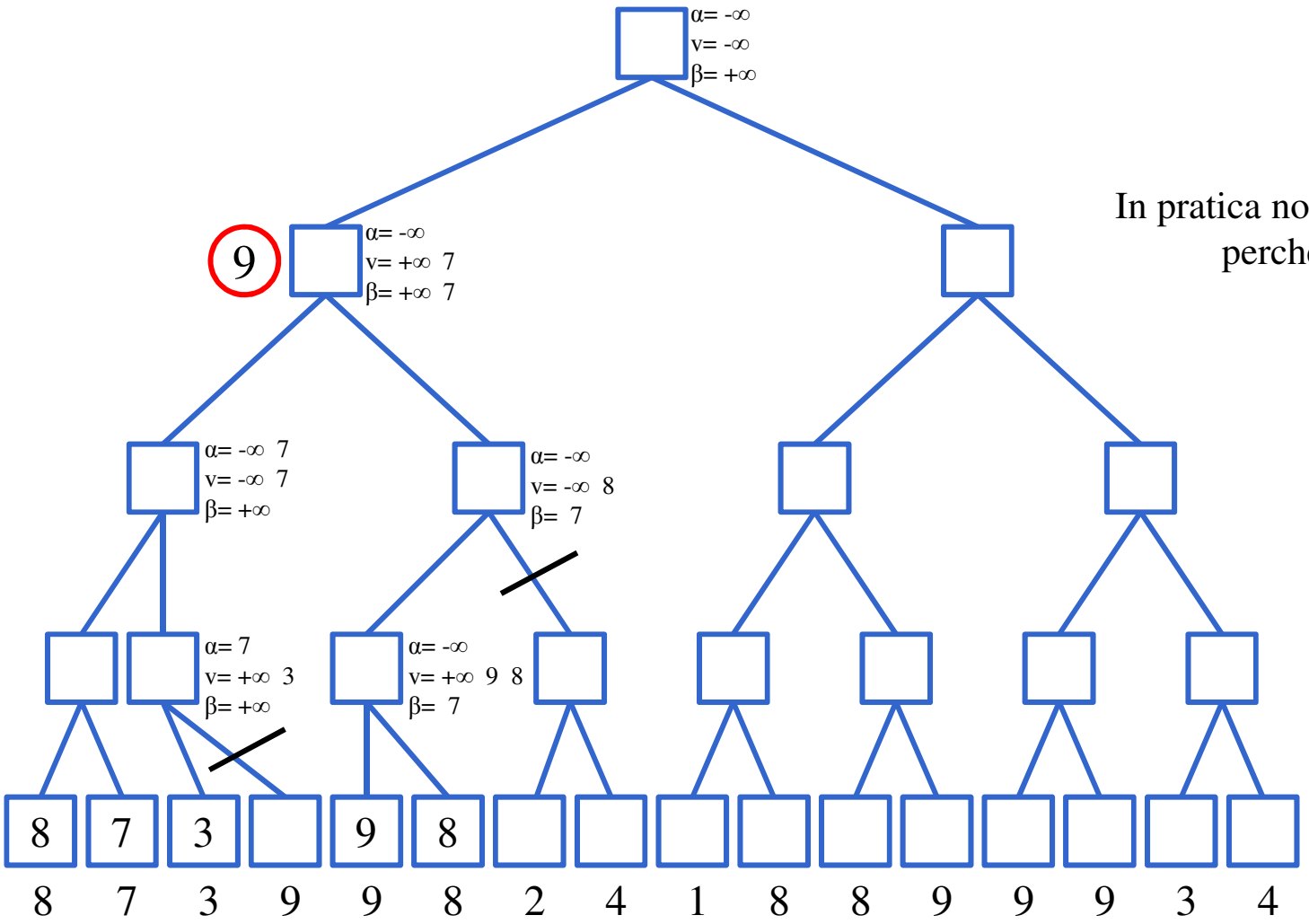
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011

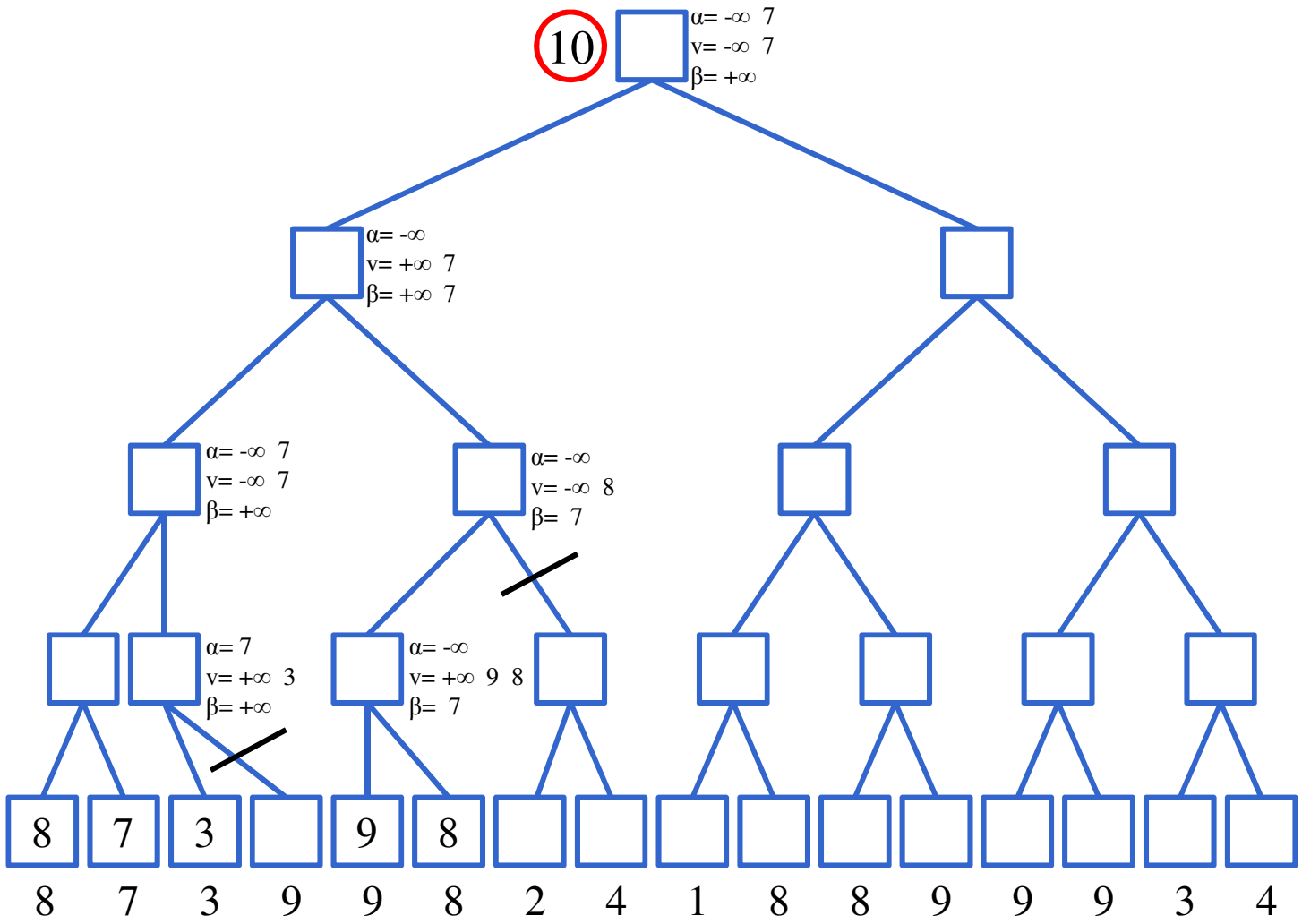


Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011

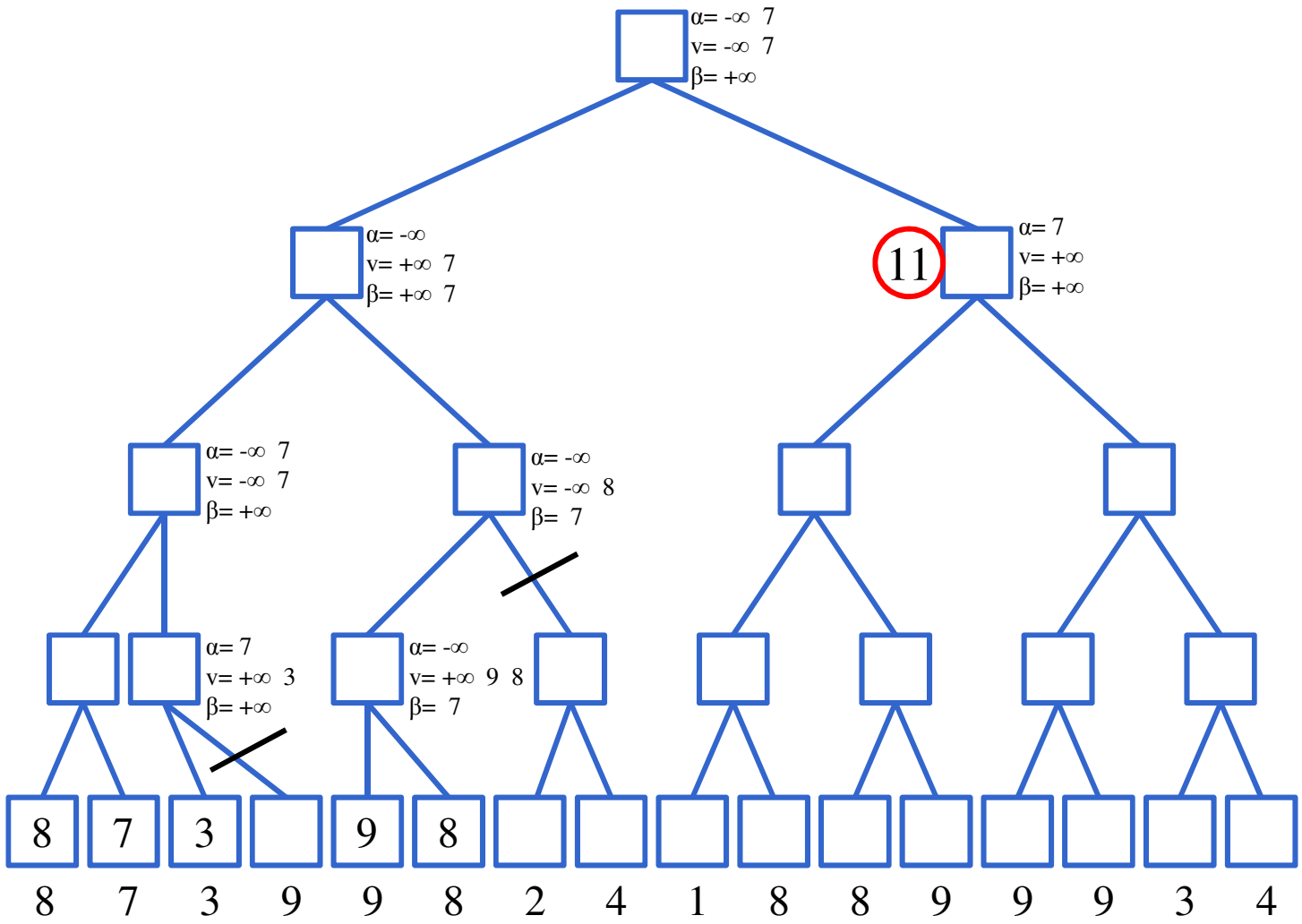


In pratica non accade nulla, perché 8>7…

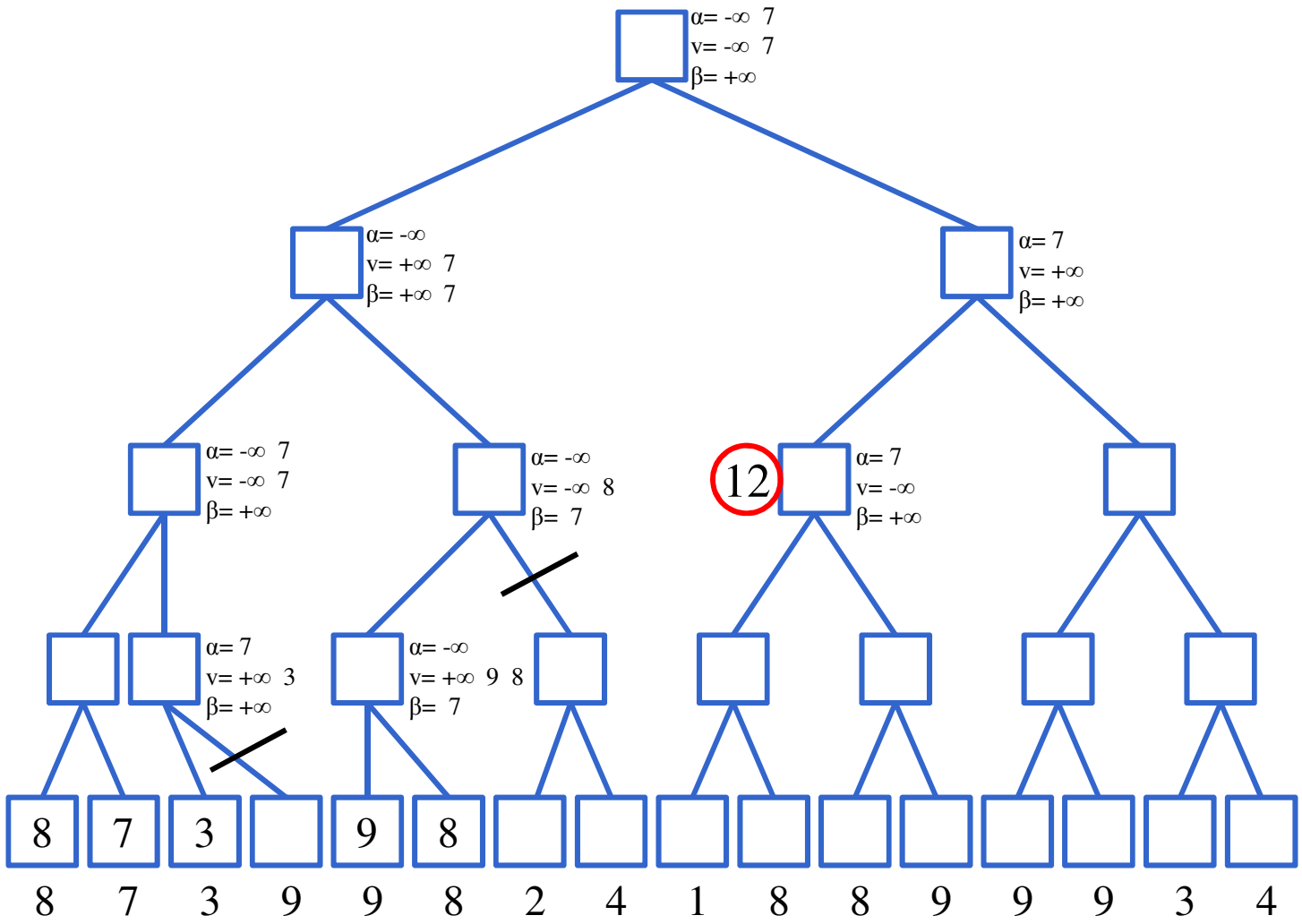
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



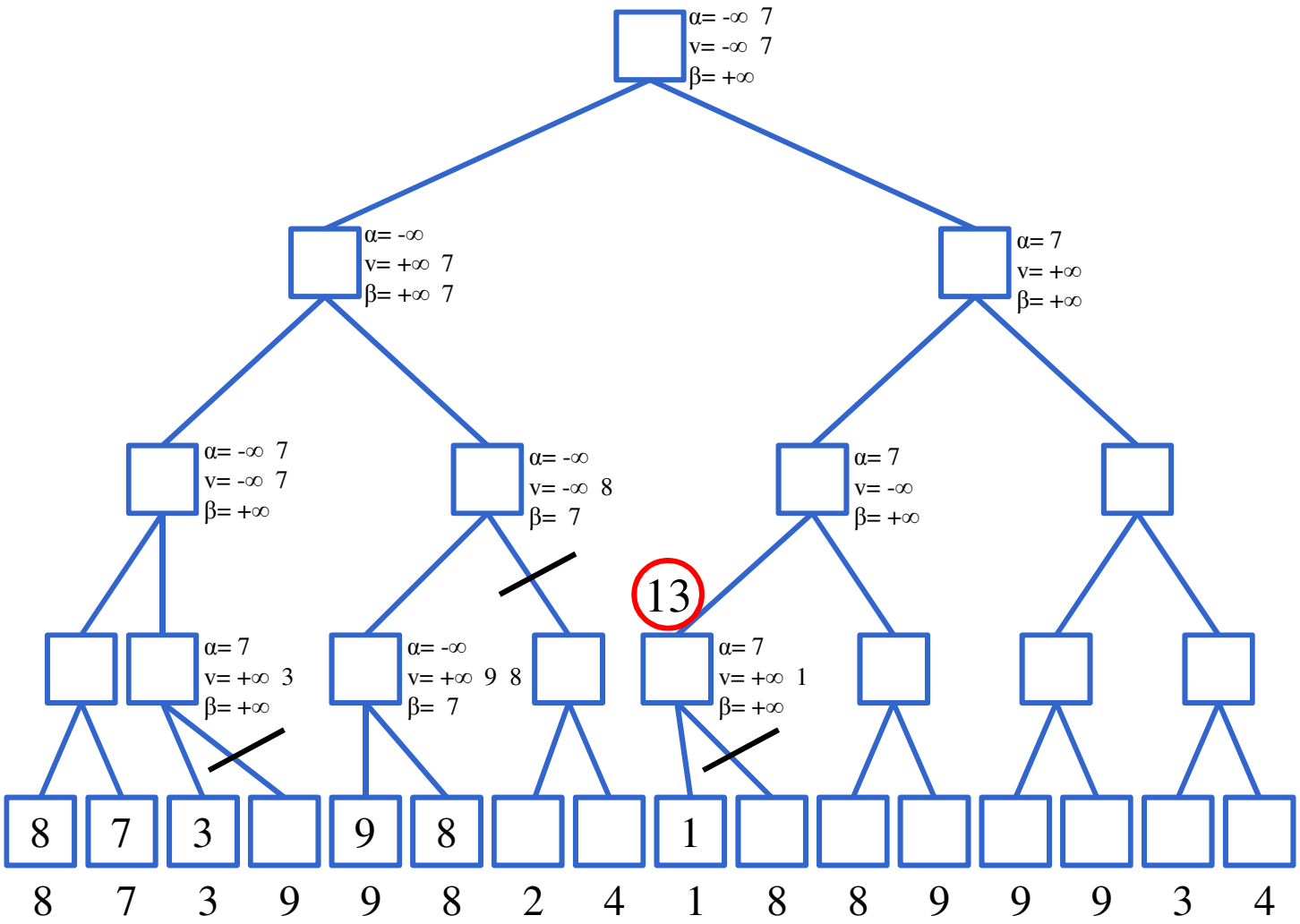
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



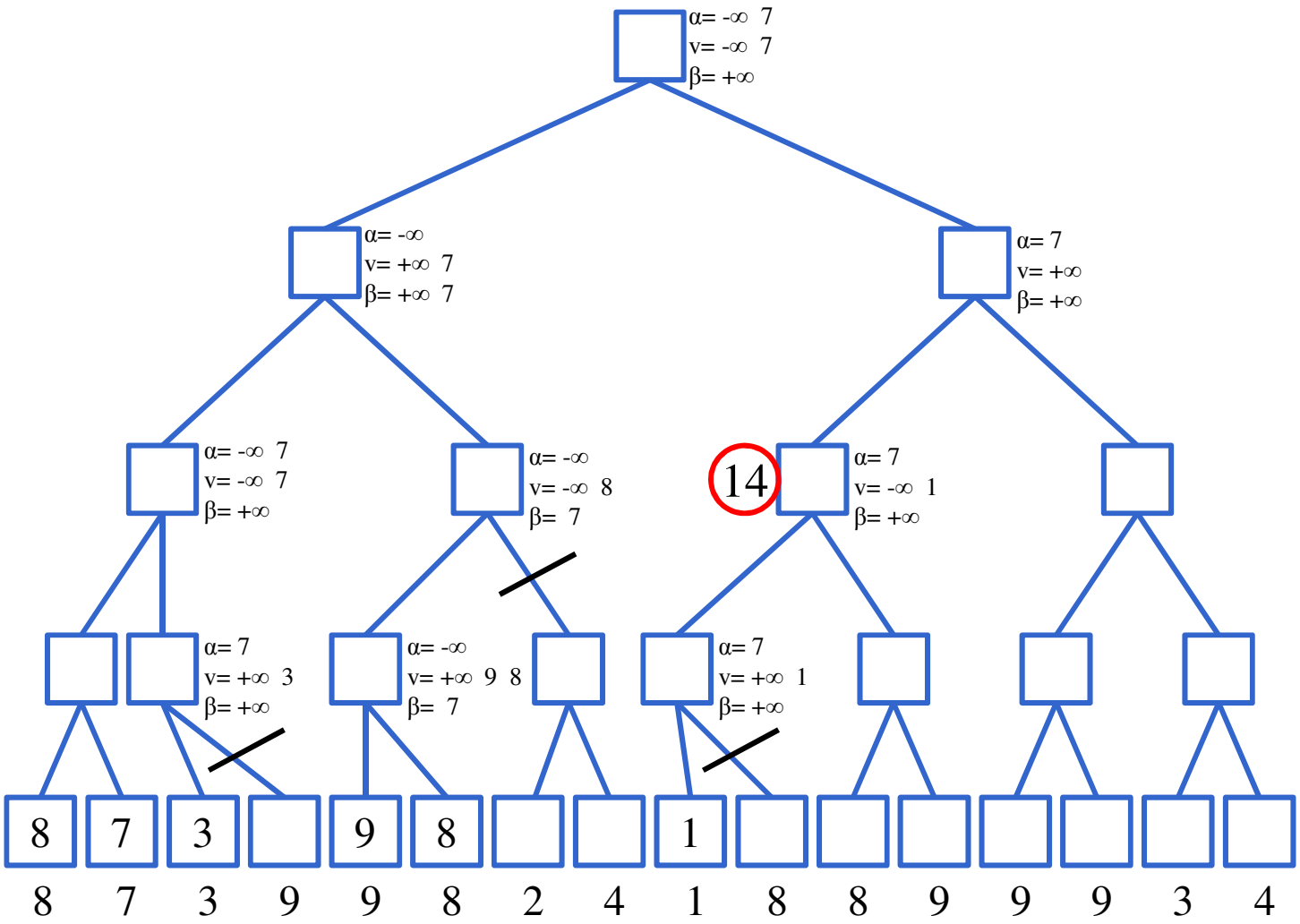
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



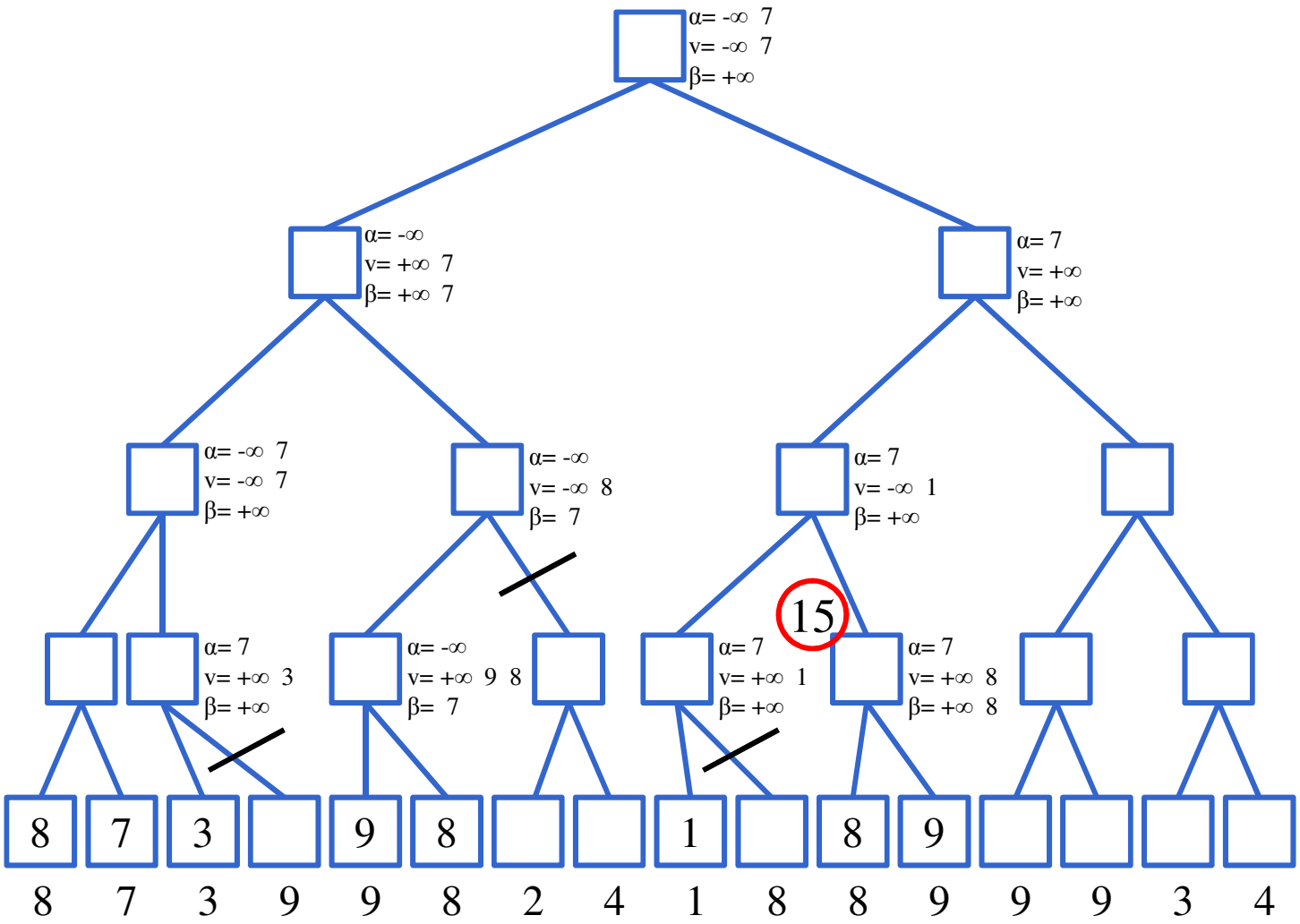
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



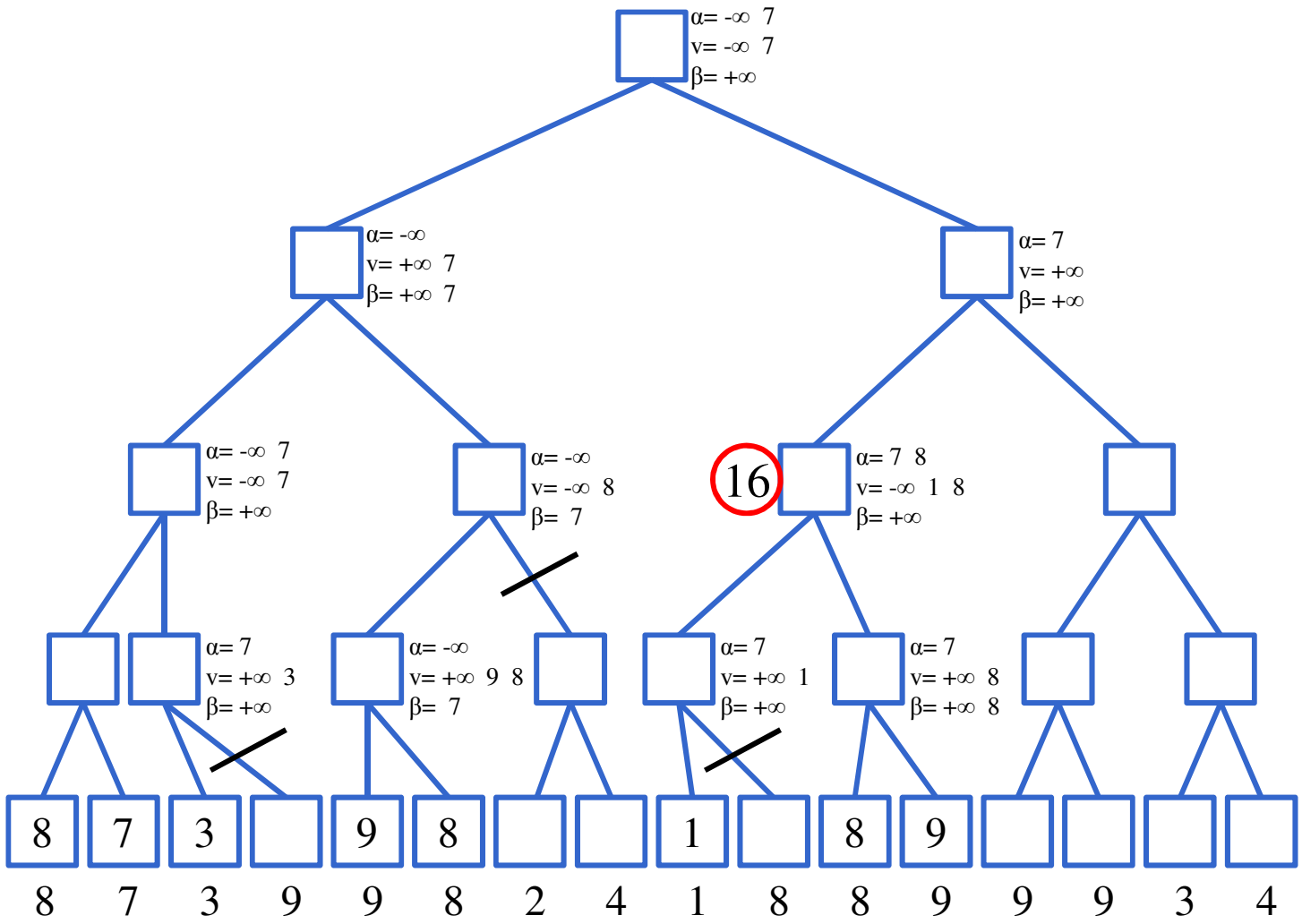
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



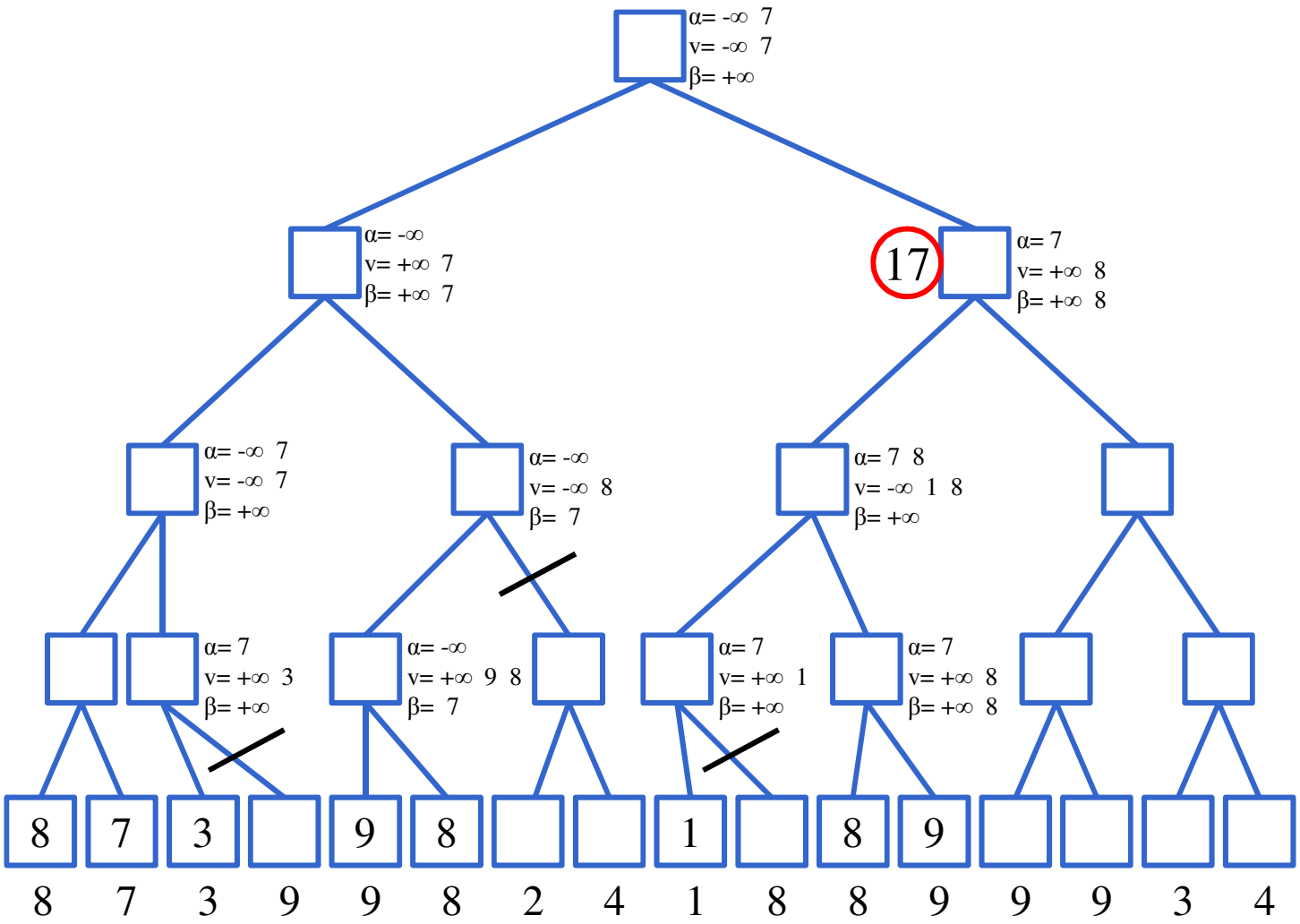
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



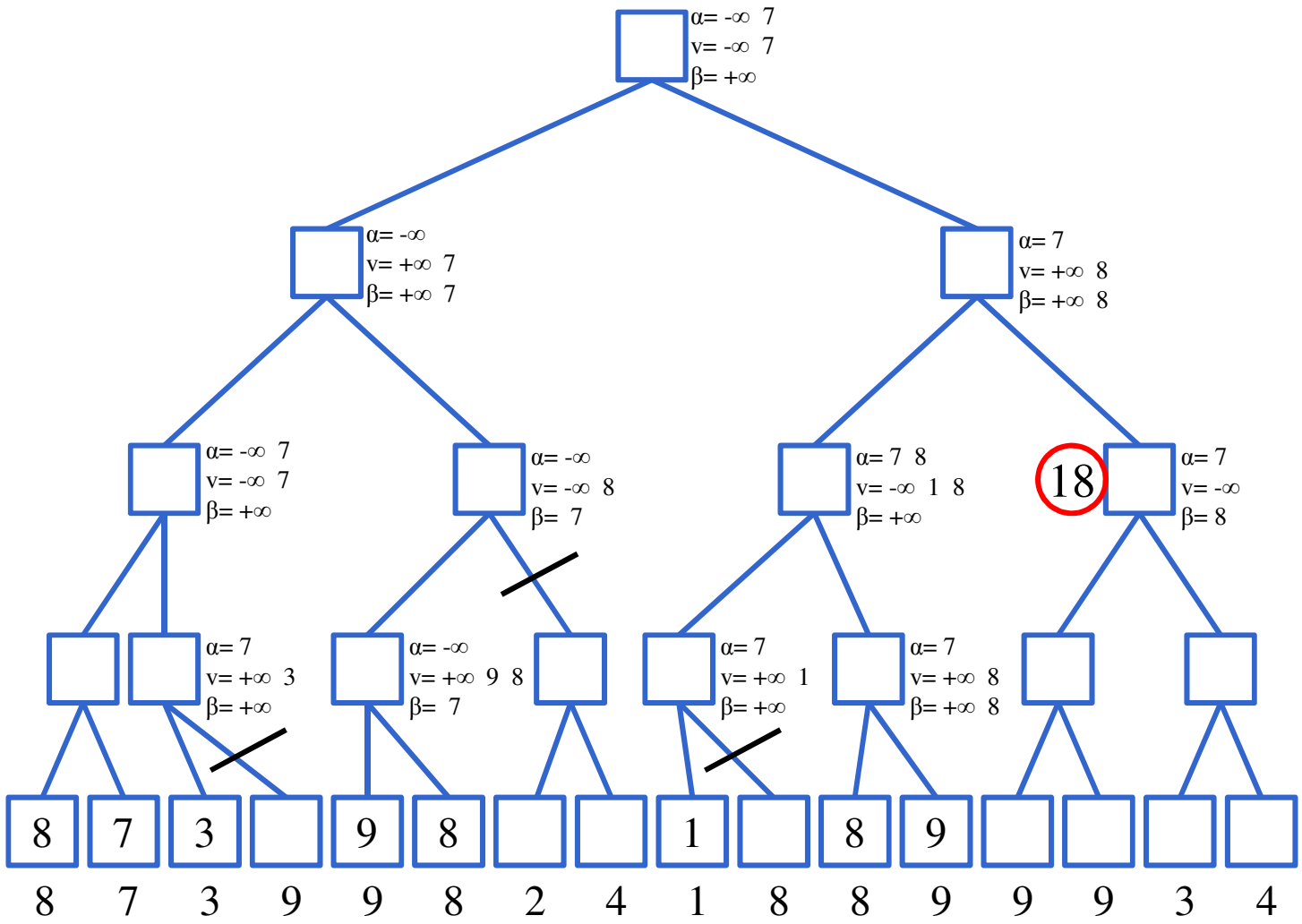
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



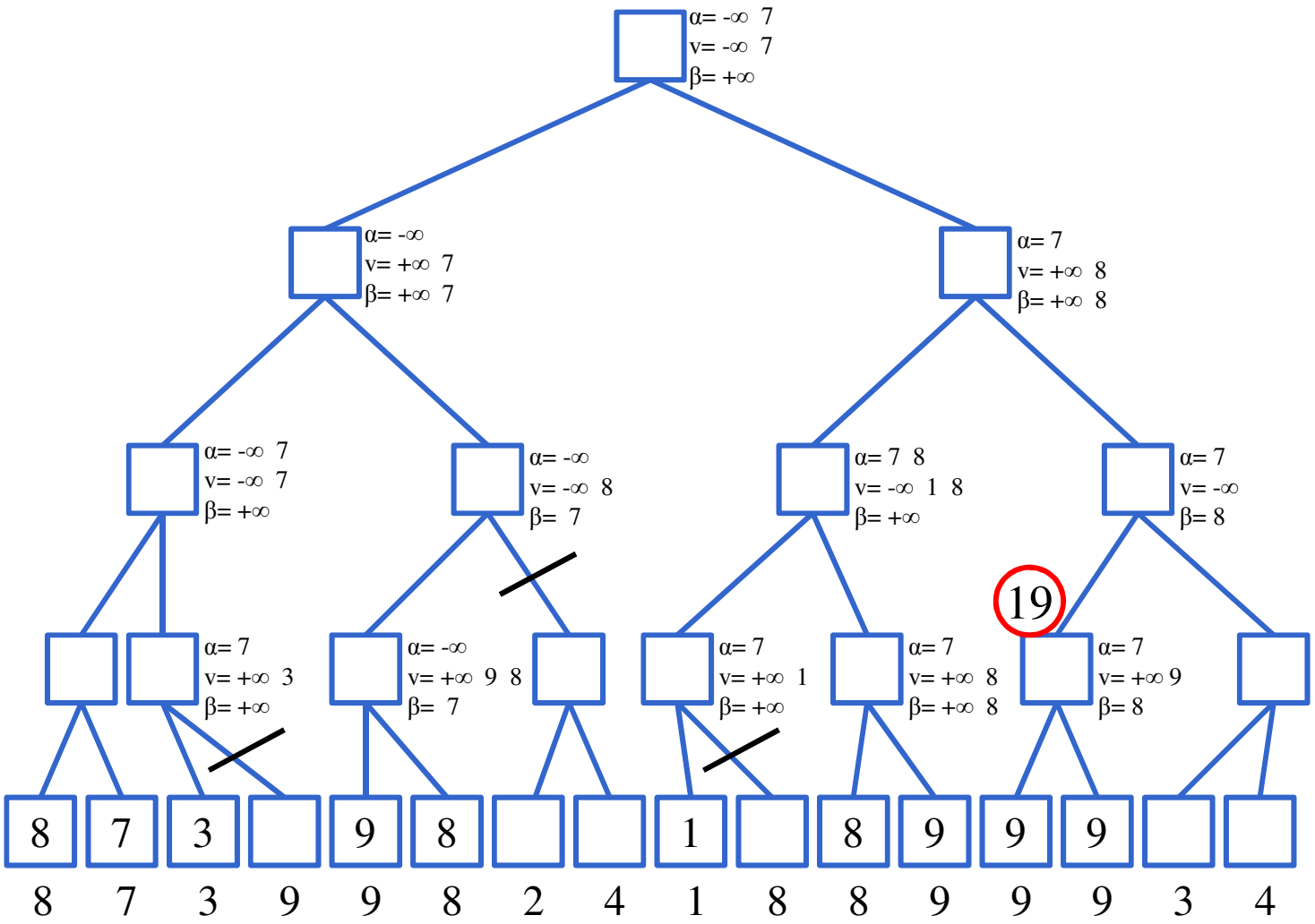
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



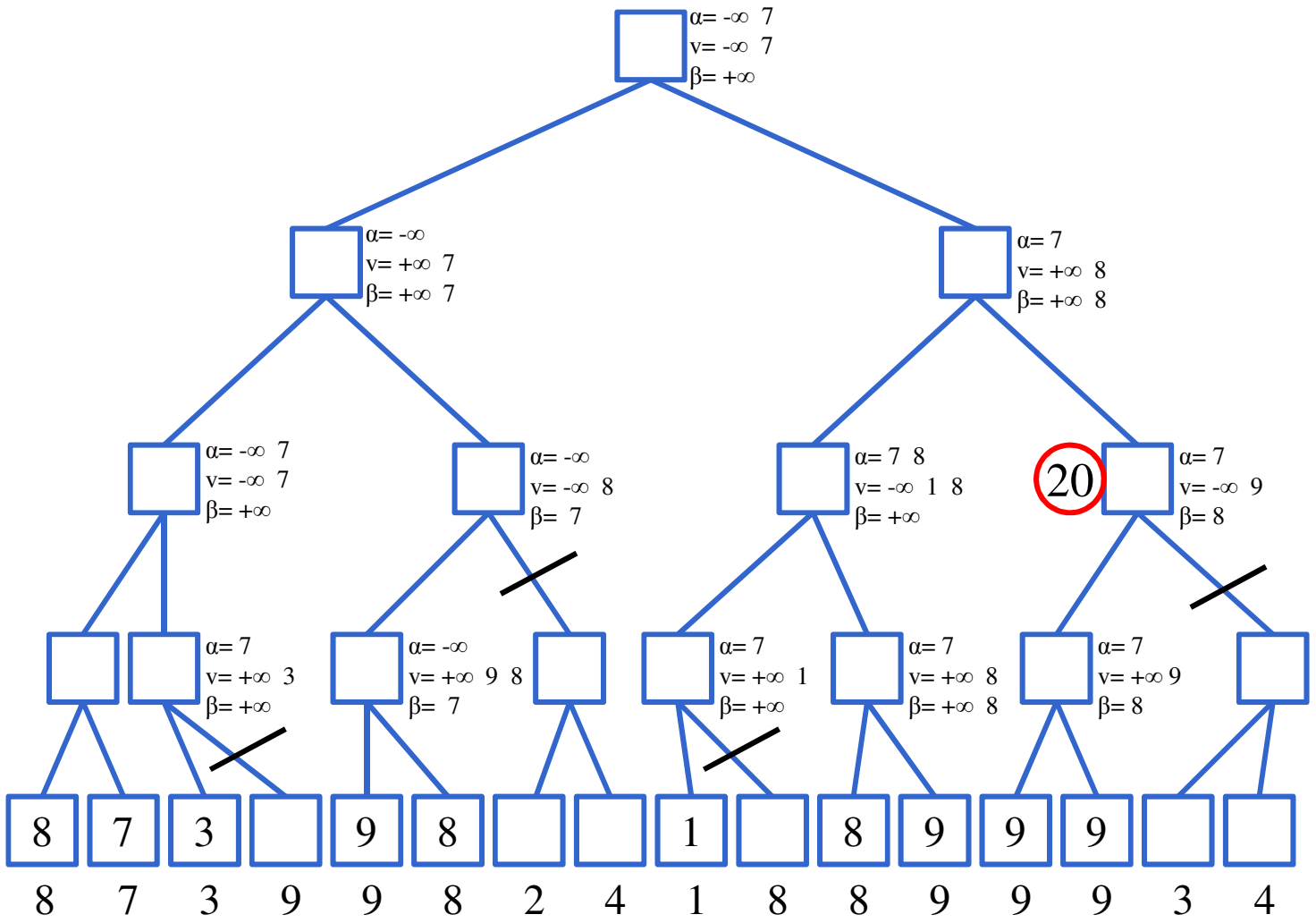
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



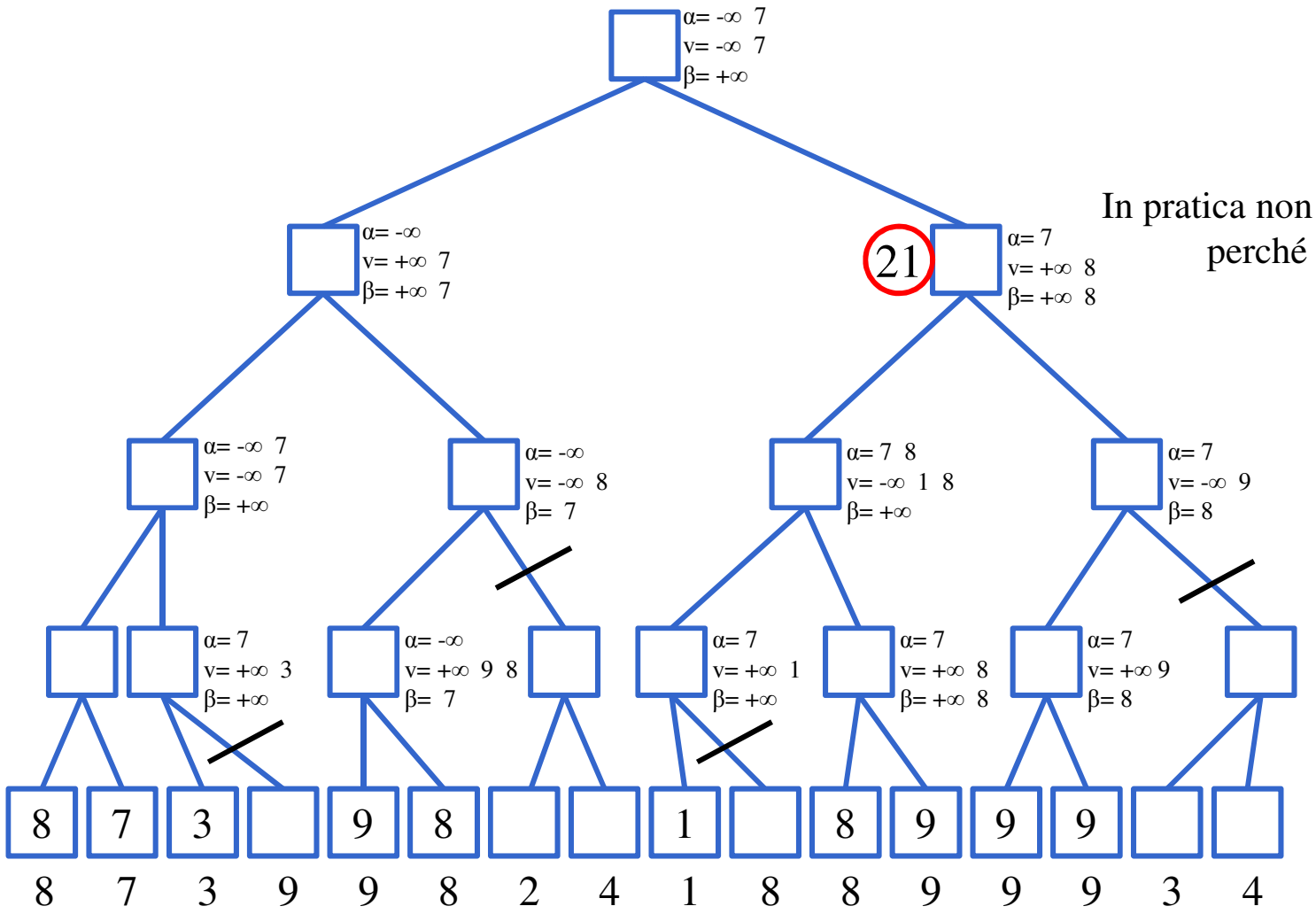
Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011

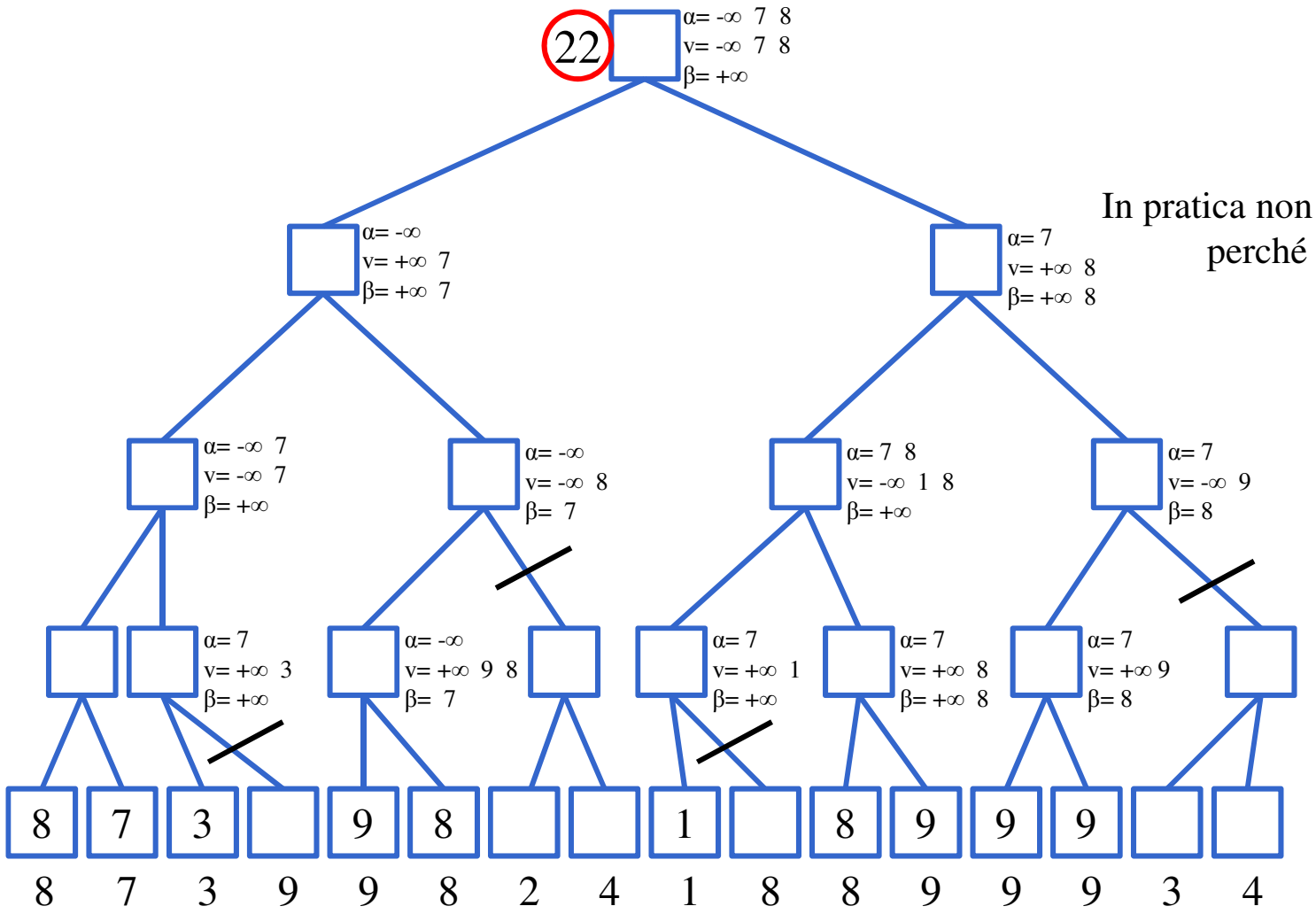


Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



In pratica non accade nulla, perché 9>7…

Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011



In pratica non accade nulla, perché 9>7…

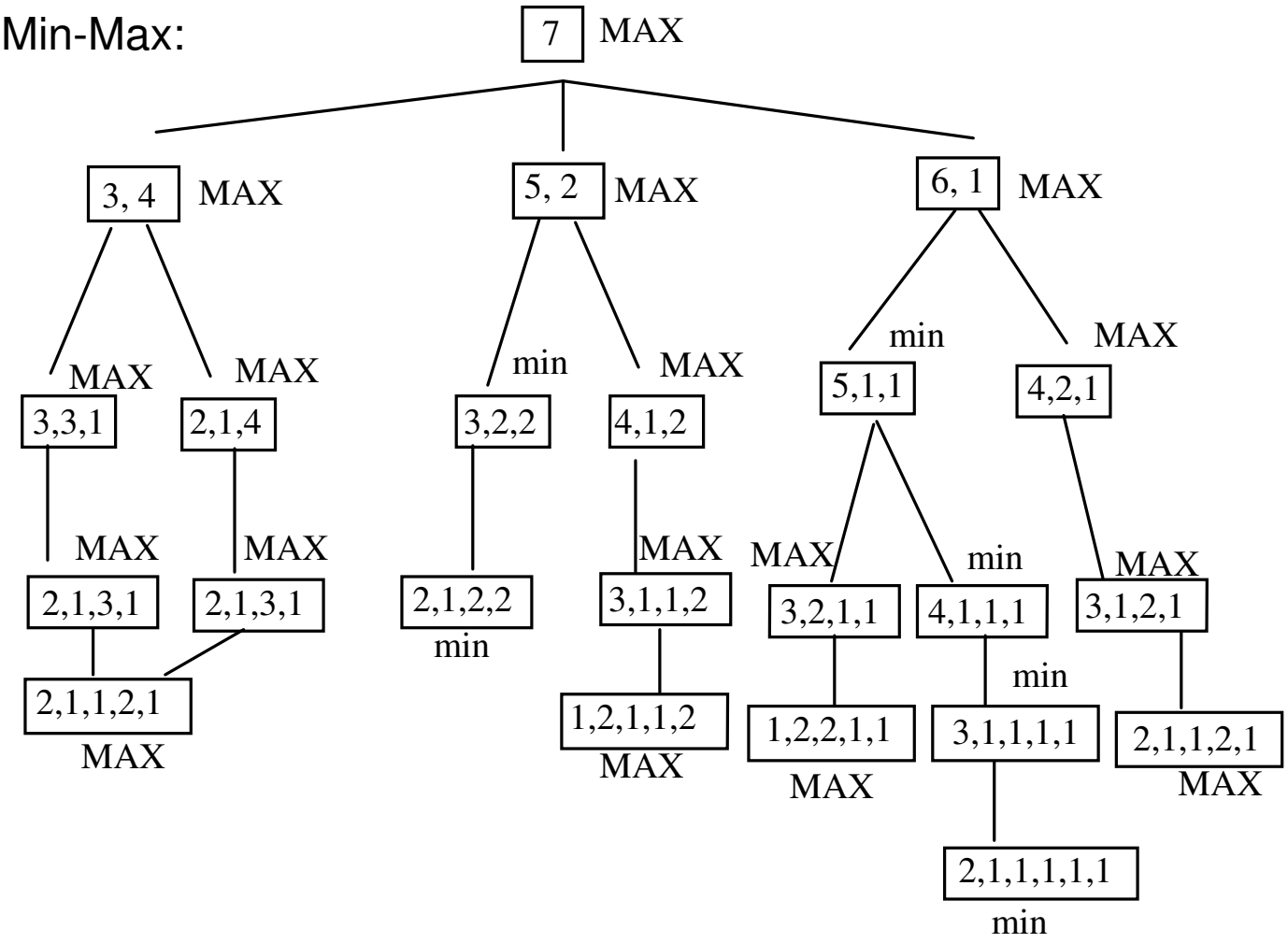
Esercizio – min-max e tagli alfa-beta

- Si consideri il seguente gioco:
- due giocatori hanno davanti una singola pila di oggetti (ad esempio una pila di monete).
- Il primo giocatore divide la pila in due pile separate che devono essere diverse. Poi continua il secondo lavorando su una a scelta delle due pile e così via.
- Il gioco termina solo quando in ogni pila ci sono una o due monete. A questo punto il primo giocatore che non può muovere ha perso.

- Si applichi la procedura MIN-MAX e si mostri lo spazio di ricerca nel caso in cui cominci MIN con una pila di 7 monete.
- Si commenti lo spazio risultante.
- Notazione: Ogni nodo dello spazio di ricerca rappresenta una configurazione: in particolare i numeri X,Y,Z indicano che si hanno tre pile corrispondenti ciascuna ad un numero, che indica il numero di monete contenute

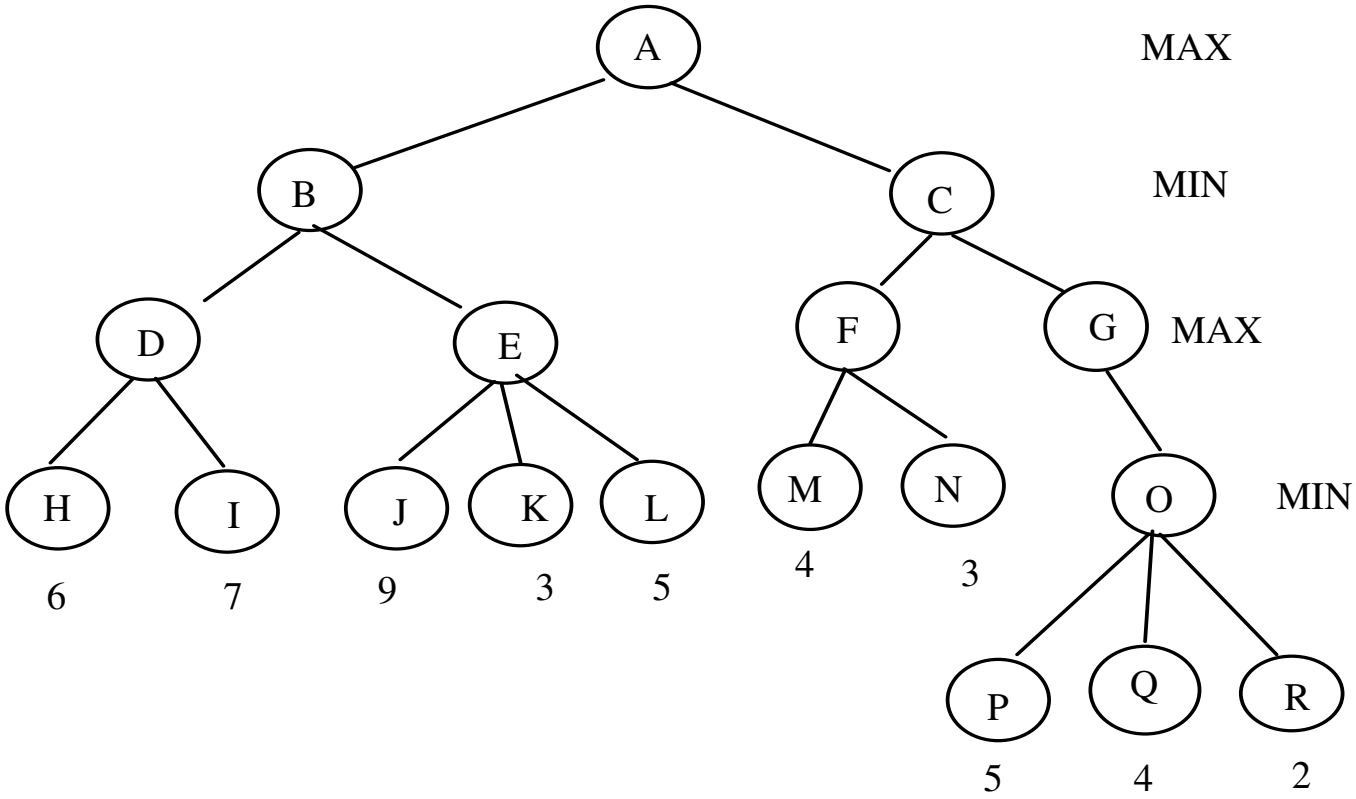
Soluzione – min-max e tagli alfa-beta

- Min-Max:



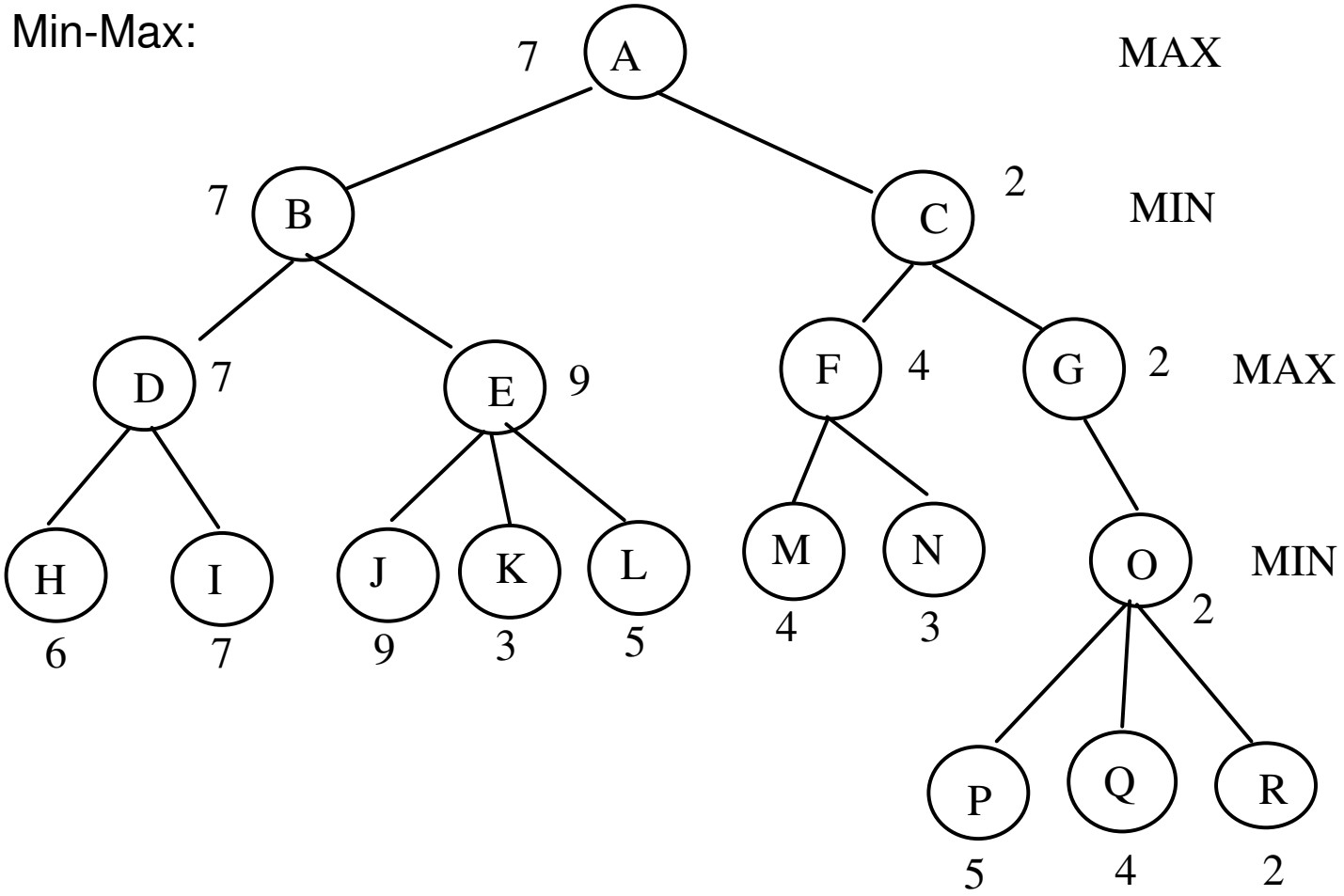
Esercizio – min-max e tagli alfa-beta

- Dato il seguente albero di ricerca per un gioco a due giocatori, si mostri quale mossa selezionerà MAX, secondo l'algoritmo MIN-MAX. Si mostri inoltre come può essere potato l'albero applicando i tagli alfa-beta.



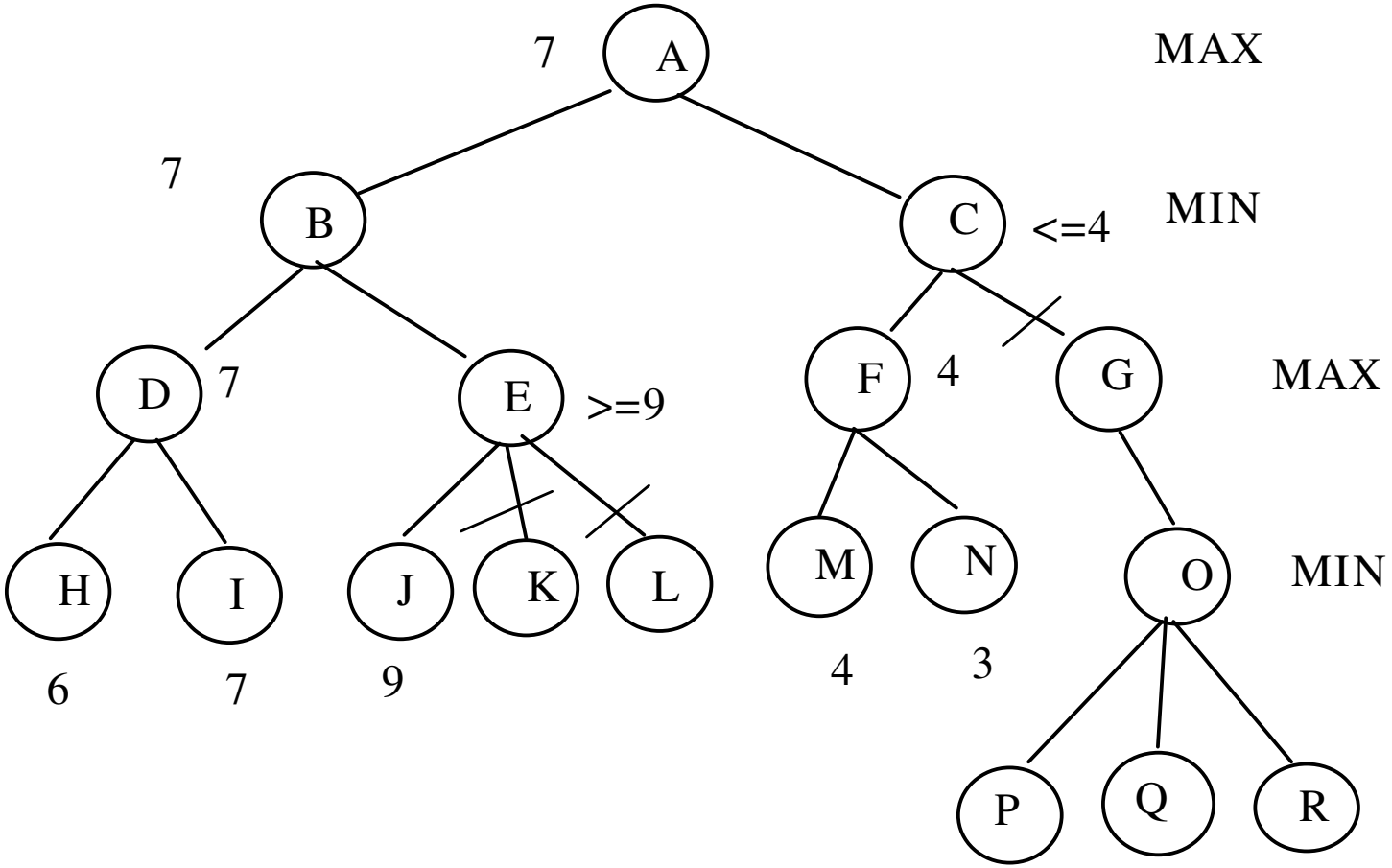
Soluzione – min-max e tagli alfa-beta

- Min-Max:



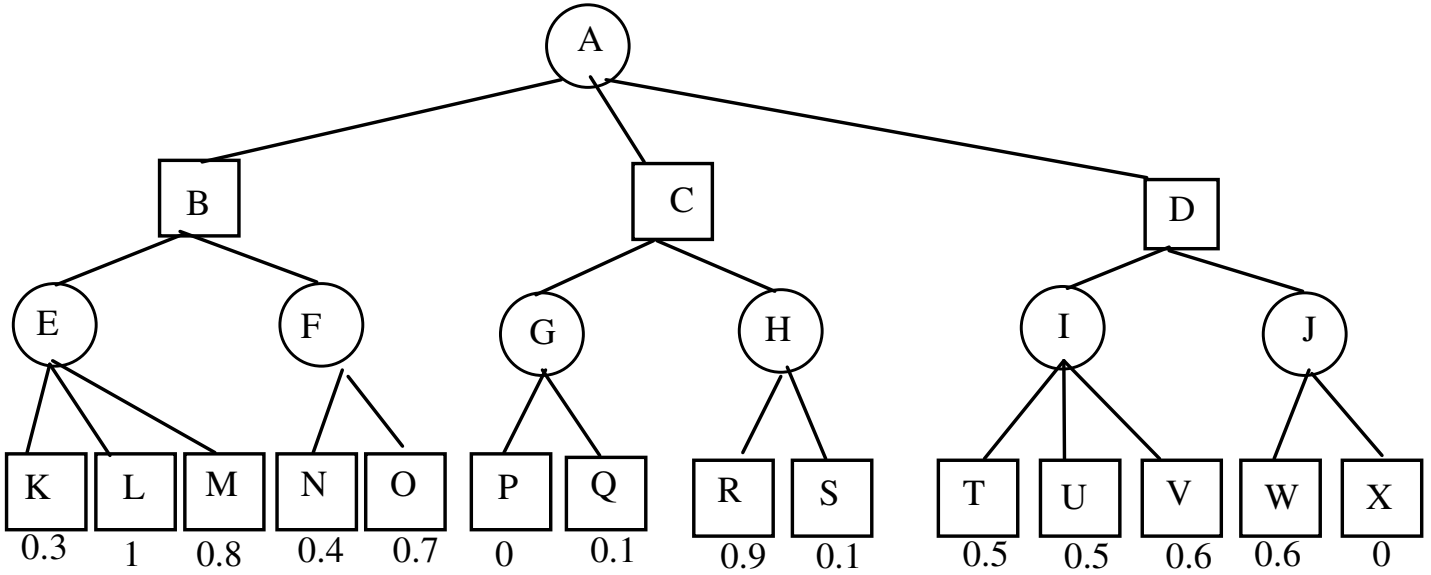
Soluzione – min-max e tagli alfa-beta

- Tagli alfa-beta:



Esercizio – min-max e tagli alfa-beta
Compito del 5 Novembre 2003

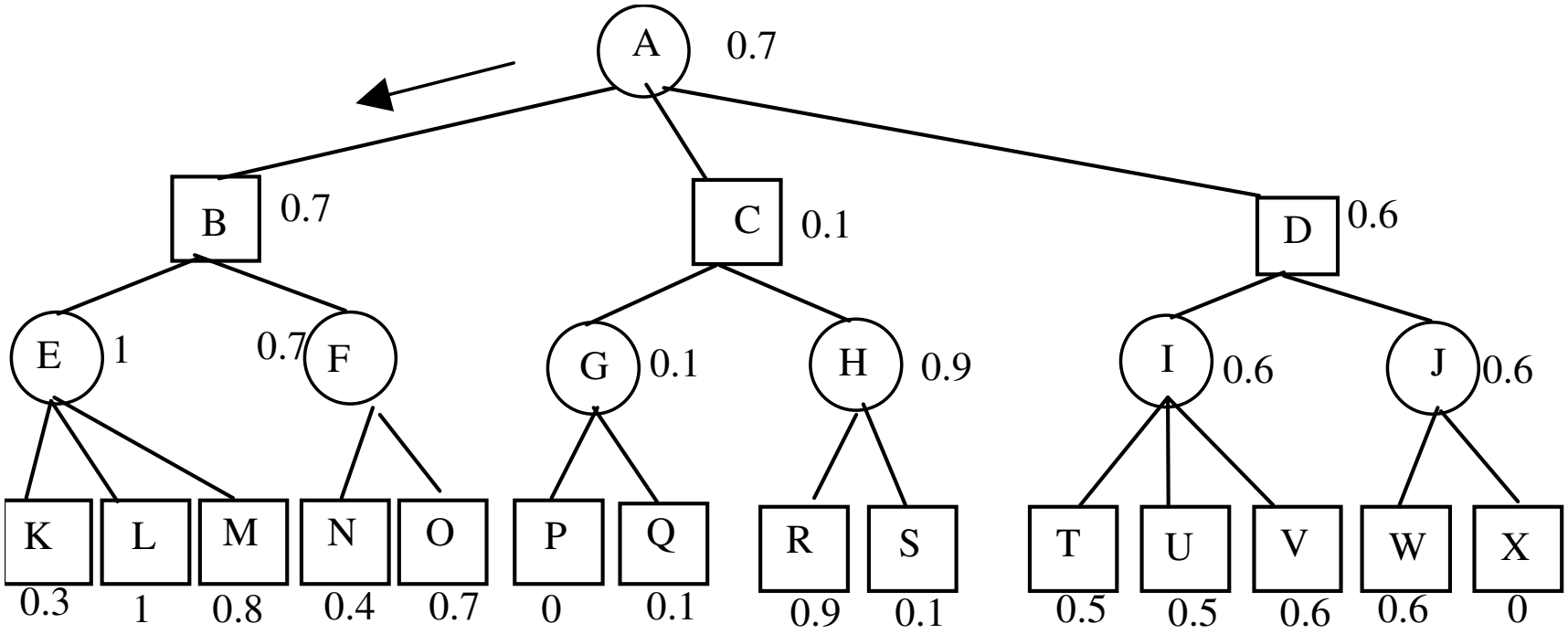
- Si consideri il seguente albero di gioco dove i punteggi (compresi tra 0 e 1) sono tutti dal punto di vista del primo giocatore (0 indica la sconfitta di MAX e 1 la sua vittoria) :



- Supponendo che il primo giocatore sia MAX, quale mossa dovrebbe scegliere? Si risolva l'esercizio tramite l'algoritmo MIN-MAX. Successivamente si mostrino i tagli alfa-beta.

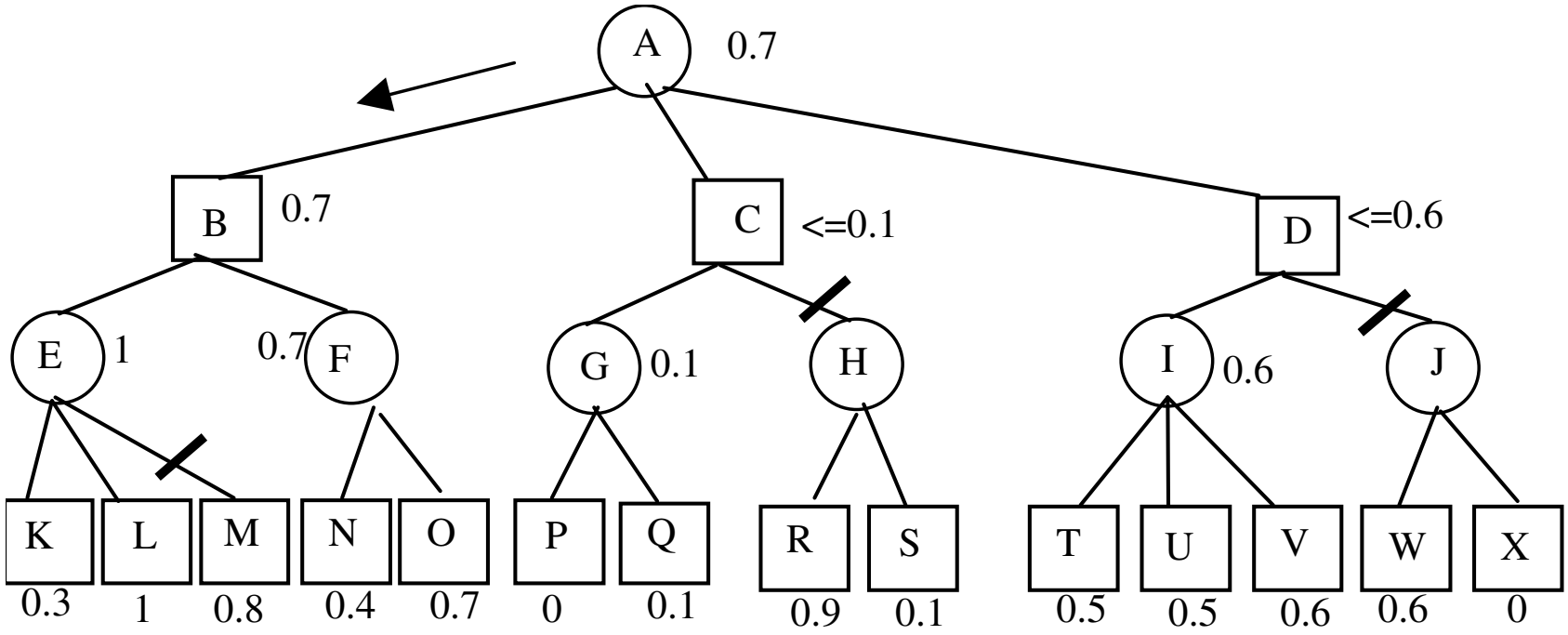
Soluzione – min-max e tagli alfa-beta

- Min-Max:



Soluzione – min-max e tagli alfa-beta

- Tagli alfa-beta:



Compito del 9 Settembre 2003 – Es. 4

Si scriva un programma Prolog **`liststruct(Lin,Lout`**) che data una lista di strutture con funtore principale **`f`** e arità **`1`** annidate a diversi livelli, restituisca in uscita una lista di interi che corrispondono alla profonfità (grado di annidamento) di ciascun elemento. La profondità **`0`** (il grado di annidamento **`0`**) e' determinato dalla costante **`f`** con arità **`0`**.

Esempio:

```
?- liststruct([f(f(f)),f, f(f)], X).
```

darà come risposta:

```
Yes X=[2,0,1]
```

Compito del 9 Settembre 2003 – Es. 4
Soluzione

```
liststruct([],[]).
liststruct([X|Y],[N|Tail]):-
      nesting(X,N), listruct(Y,Tail).

nesting(f,0).
nesting(f(X),N):-
      nesting(X,N1), N is N1 +1.
```

Compito del 5 Novembre 2003 – Es. 2

Si definisca un predicato **`no_ripetuti(Xs, Ys,N`**) che è vero se **`Ys`** è la lista degli elementi in **`Xs`** senza duplicazioni e **`N`** il numero di elementi ripetuti.

Esempio:

```
:- no_ripetuti([3,5,3,9,9,8,9], Ys, N).
yes
Ys=[5,3,8,9] N=3
```

Compito del 5 Novembre 2003 – Es. 2 Soluzione

```
no_ripetuti([], [], 0).
no_ripetuti([X|Xs], Ys, N):-
  member(X, Xs),
  no_ripetuti(Xs, Ys, N1), N is N1 + 1.
no_ripetuti([X|Xs], [X|Ys], N):-
  not member(X, Xs),
  no_ripetuti(Xs, Ys, N).

member(X,[X|Xs]).
member(X,[Y|Ys]):- member(X,Ys).
```

Compito del 9 Dicembre 2003 – Es. 3

Si scriva un programma Prolog **`listindex(Lin1,Lin2,Lout)`** che date due liste di elementi **`Lin1`** e **`Lin2`** contenenti numeri , produca una lista in uscita **`Lout`** contenente come elementi solo gli elementi della lista **`Lin2`** di posizione specificata da **`Lin1`**. Si noti che **`Lin1`** contiene sempre numeri >=1. Si scrivano esplicitamente tutti i predicati Prolog usati nella soluzione.

Esempio:

```
listindex([5,3,1], [9,20,11,5,19,21,0], X)
```

restituisce

```
X=[19, 11, 9]
```

Compito del 9 Dicembre 2003 – Es. 3
Soluzione

```
listindex([], L, []).
listlindex([E1|Rest1], L2, [N3|Rest3]):-
      estrai(E1, L2, N3),
      listindex(Rest1, L2, Rest3).

estrai(1,[A|B],A):- !.
estrai(Nin,[A|B],Nout):-
      N is Nin - 1, estrai(N,B,Nout).
```

Compito del 21 Giugno 2005 – Es. 1 (6 punti)

Si traducano in logica dei predicati del primo ordine le seguenti frasi:

A. *Se qualcuno risparmia del denaro guadagna un certo interesse.*
B. *Se non ci fosse un interesse nessuno risparmierebbe denaro.*

Dimostrare con il metodo di risoluzione che *B* è conseguenza logica di *A.*
Si utilizzi nella formalizzazione il seguente vocabolario:

- rd(X, Y): X risparmia la somma di denaro Y
- gi(X, Z): X guadagna l'interesse Z

Compito del 21 Giugno 2005 – Es. 2 (6 punti)

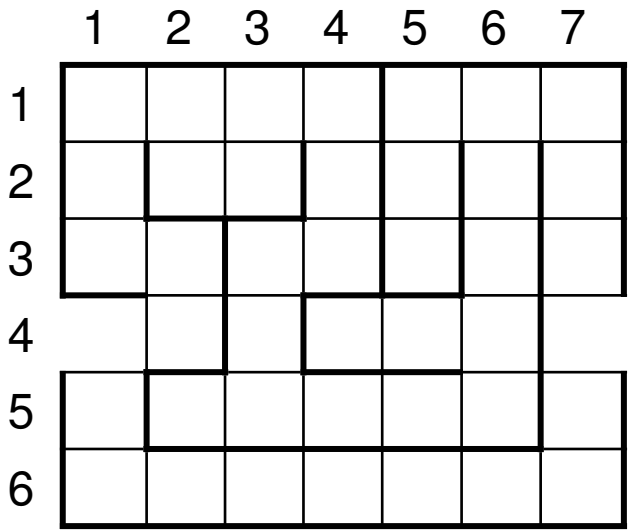
Si scriva un programma PROLOG che data in ingresso una lista di liste **`Lin`**, dia in uscita una nuova lista di liste **`Lout`** in cui ogni lista interna non condivide elementi con la precedente lista interna. Si supponga per semplicità che le liste interne non possiedano elementi ripetuti. Si definiscano tutti i predicati utilizzati.

- Esempio:

```
?-reduce([[1,5,2,4,9], [3,7,2], [9,3]],Z).
Z = [[1,5,2,4,9], [3,7], [9]]
```

Compito del 21 Giugno 2005 – Es. 3 (10 punti)

Dato il seguente labirinto rappresentato su una griglia 6x7:



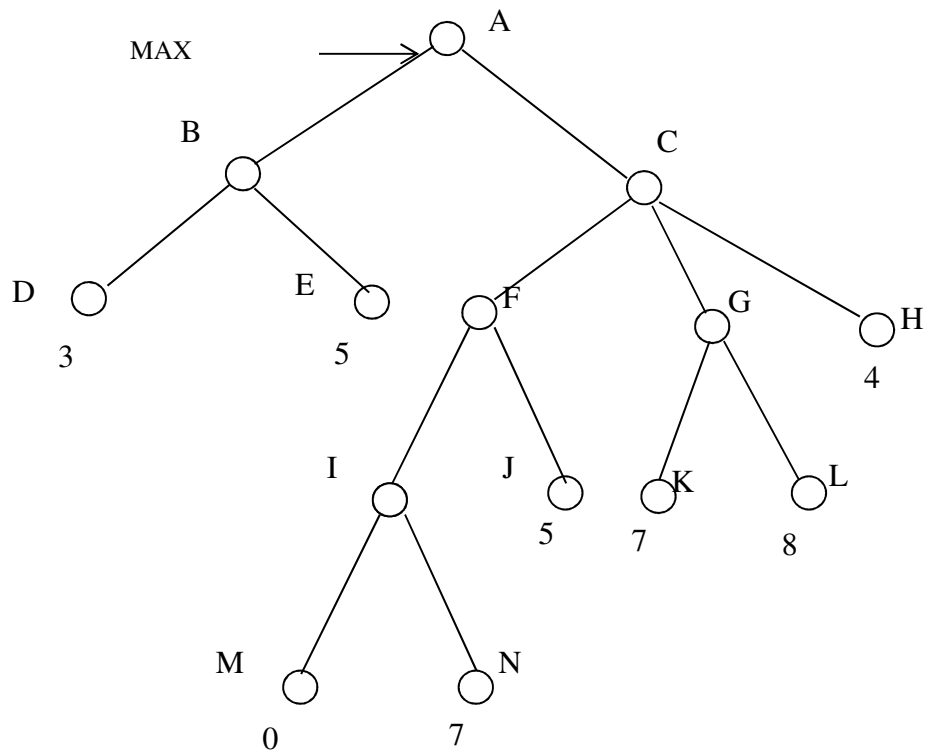
a. formalizzare il problema di trovare un percorso dall'ingresso all'uscita come un problema di ricerca in uno spazio di stati;

b. descrivere un algoritmo di tipo A* per i grafi per trovare la soluzione; *Suggerimento: si scelga come stima la Manhattan distance tra due punti (x1,y1), (x2,y2) che è data dalla formula: M = |y2 - y1 | + |x2 - x1|*

c. mostrare come si comporta l'algoritmo (mediante l'albero di ricerca). Trova la soluzione ottimale?

Compito del 21 Giugno 2005 – Es. 4 (7 punti)

Sia dato il seguente esempio di gioco con avversario in cui si assume una funzione di valutazione degli stati terminali che valuta le foglie come indicato in figura:



1. Evidenziare le valutazioni dei nodi mediante l'algoritmo MIN MAX e dire quale mossa farebbe MAX.
2. Tracciare l'esecuzione dell'algoritmo con potatura α-β con visita dell'albero da sinistra a destra (evidenziando le potature). Tracciare l'esecuzione di α-β con visita dell'albero da destra a sinistra. La mossa scelta è la stessa nei due casi? I nodi non visitati sono gli stessi nei due casi? Perché?

Compito del 22 Giugno 2006 – Es. 1 (7 punti)

L'algoritmo di Euclide per il calcolo del massimo comun divisore è dato dalla seguente definizione:

$$mcd(A,B) = \begin{cases} A & se \quad A = B \\ mcd(A, B-A) & se \quad A < B \\ mcd(A-B, B) & se \quad A > B \end{cases}$$

Si consideri il seguente programma Prolog, che è l'implementazione dell'algoritmo di Euclide

```
mcd(A,A,A):-!.
mcd(A,B,M):- A<B,!, C is B-A, mcd(A,C,M).
mcd(A,B,M):- I is A-B, mcd(I,B,M).
```

Sebbene sembri corretto, l'algoritmo va in loop con il goal

```
?- mcd(2,2,1).
```

Si corregga il programma e si disegni l'albero SLD relativo all'invocazione

```
?- member(X,[1,2]), mcd(2,X,1).
```

dove il predicato member/2 è definito da

```
member(Y,[Y|T]).
member(Y,[Z|T]):- member(Y,T).
```

Compito del 22 Giugno 2006 – Es. 2 (5 punti)

Si definisca un predicato prodotto_cartesiano(L1,L2,L3) che, date due liste qualsiasi L1 e L2 restituisca la lista L3 delle coppie ordinate che si possono formare con elementi di L1 e L2.

Per esempio, se L1=[a,b,c], L2 = [a,d] la lista risultante deve essere:

L3=[[a,a],[a,d],[b,a],[b,d], [c,a],[c,d]].

Si definiscano tutti i predicati utilizzati, anche se già visti a lezione.

Compito del 22 Giugno 2006 – Es. 3 (7 punti)

Si formalizzino le seguenti frasi in logica dei predicati:

- Giorgio e Paolo amano le stesse donne
- Se Giorgio ama una donna allora è felice.
- Giorgio non è felice.

Le si trasformi in clausole, si indichi se sono tutte clausole di Horn e poi si usi poi il principio di risoluzione per derivare il teorema “Paolo non ama alcuna donna”.

Compito del 22 Giugno 2006 – Es. 4 (5 punti)

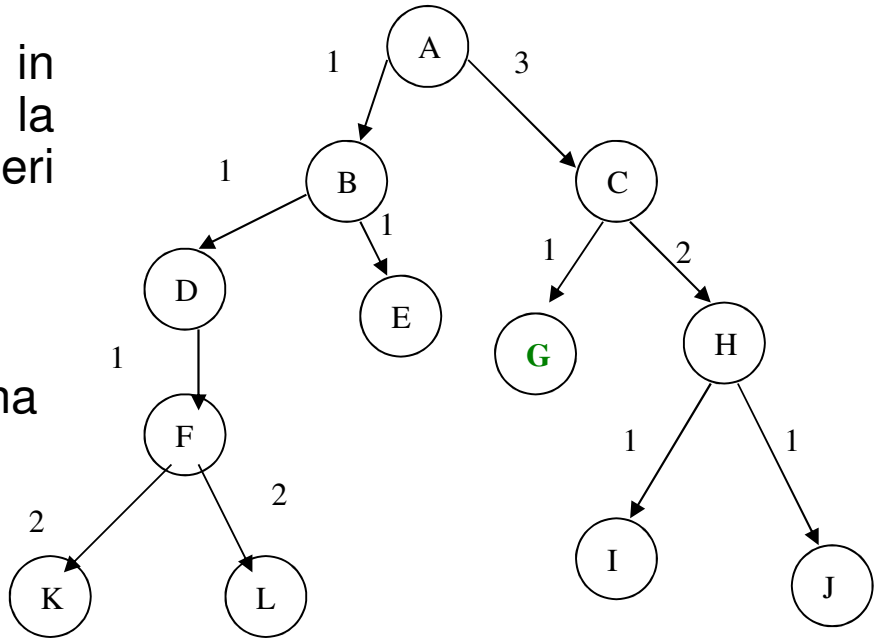
Si consideri il seguente albero di ricerca in cui i numeri sugli archi sono i costi.

Si assuma che i nodi siano espansi in ordine alfabetico, nel caso in cui la strategia di ricerca applicata li consideri equivalenti.

Il goal sia lo stato G e la ricerca termini al suo raggiungimento.

Si indichi la lista di stati espansi da ognuna delle seguenti strategie di ricerca:

- Breadth First
- Depth First
- Iterative Deepening Search
- Uniform Cost Search



Si indichi inoltre complessità Spaziale, Temporale, completezza ed ottimalità di ognuna di tali strategie, motivando la risposta.

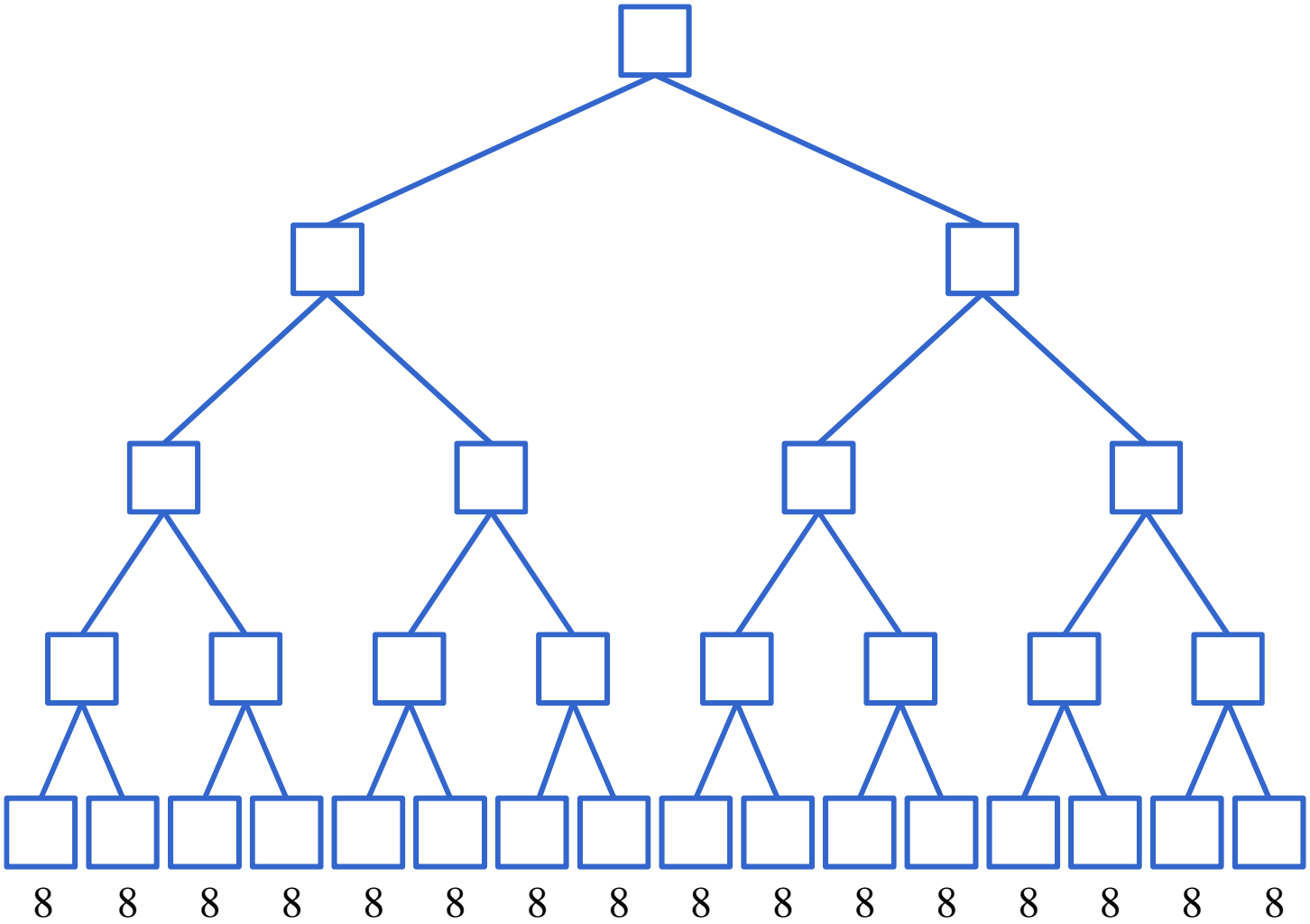
Compito del 22 Giugno 2006 – Es. 5 (8 punti)

Si modelli il seguente problema a vincoli: si hanno 6 attività di cui devo stabilire una posizione relativa rispetto alle altre. Sappiamo che:

- L'attività A non e' la prima e non è l'ultima.
- L'attività C è dopo l'attività B (non necessariamente contigue)
- L'attività E sta tra l'attività A e l'attività C
- L'attività F viene subito dopo l'attività A
- Le attività non possono sovrapporsi

Si modelli il problema usando opportune variabili domini e vincoli. Si applichi l'arc consistenza al problema originale. Si risolva il problema mostrando l'albero fino alla prima soluzione utilizzando l'arc consistenza ad ogni nodo e selezionando le variabili con l'euristica MRV(minimum remaining values). In caso di parità con euristica MRV si selezionino le variabili in ordine alfabetico, dalla prima all'ultima.

Esercizio – tagli alfa-beta con algoritmo



Esercizio – tagli alfa-beta con algoritmo Esame del 21/12/2011