

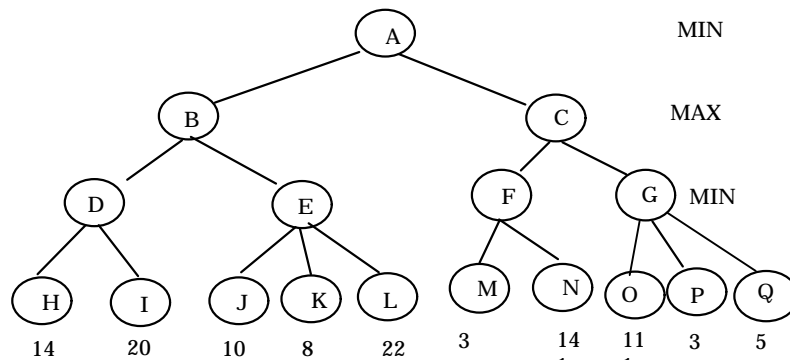
Domande sui tagli Alfa-Beta

Nota: Lo studente che ha posto queste domande, le ha presentate contemporaneamente a due persone diverse ed ha quindi ricevuto due possibili spiegazioni. Abbiamo raccolto qui entrambe le risposte, rappresentate con font e colori diversi, sperando che "ripetere giovi".

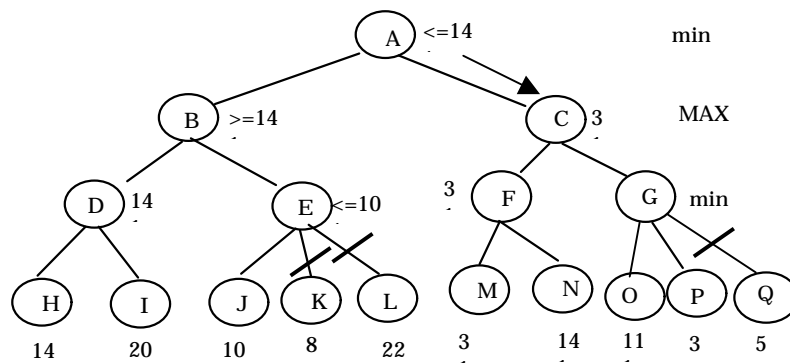
Risposta 1:

Uno stesso albero di ricerca può essere tagliato, secondo l'algoritmo alfa beta in più modi oppure la soluzione è UNICA? E cioè, con riferimento per esempio all'esercizio 3 del 11/2/2000:

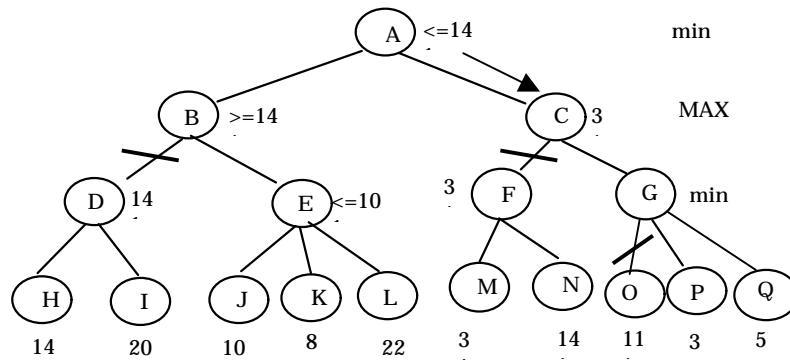
Dato il seguente albero di ricerca per un gioco a due giocatori, si mostri quale mossa selezionerà MIN, secondo l'algoritmo MIN-MAX. Si mostri inoltre come può essere ridotto l'albero applicando i tagli alfa-beta.



Oltre alla soluzione proposta:



**Potrebbe essere corretta anche questa?



NO, la soluzione è una

**E se no perché?

NEL MIN-MAX ESPLORIAMO COMUNQUE L'ALBERO IN PROFONDITA' e da sinistra a destra, aggiornando le etichette dei nodi genitore e antenati via via. Ogni ramo rappresenta una mossa e le foglie il punteggio raggiunto. La mossa scelta dal primo genitore è quindi determinata in base all'esplorazione dell'albero e all'aggiornamento delle etichette dei nodi radice e intermedi.

Per esempio, l'etichetta del nodo E sarà prima 10 considerando il nodo J figlio di E, poi diventa 8 toccando il nodo K e tale resta anche dopo aver considerato il nodo L (gioca min e 22 è un punteggio maggiore di 8 e quindi penalizzante per min che tende a fare la mossa a più basso punteggio).

L'algoritmo è simile se si considerano i tagli alfa-beta, con l'aggiunta di possibili tagli nell'esplorazione dell'albero (ovvero parti dell'albero non vengono considerate) determinati dal valore corrente delle etichette dei nodi genitore (detti valore alfa e valore beta a seconda che il nodo sia max o min). Inizialmente i nodi min antenati hanno un'etichetta che vale più infinito e i nodi max antenati hanno etichetta meno infinito.

Per esempio, fino a che non tocco il nodo H (la prima foglia toccata di fatto nell'esplorazione dell'albero) i nodi min antenati hanno un'etichetta che vale più infinito e non posso operare alcun taglio perché nessun valore può essere maggiore o uguale) e i nodi max antenati hanno etichetta meno infinito (nessun taglio perché nessun valore può essere minore o uguale).

Quello che non va nella sua soluzione è che arrivati al nodo B non sono in grado di operare alcun taglio, perché non ho ancora toccato alcuna foglia e non conosco il punteggio che si ottiene arrivando a quel punto del gioco.

**Perché nella soluzione da voi proposta nelle foglie 11, 3 e 5 si è scelto di non espandere 5 invece di 11 che è meno interessante per MIN?

QUANDO si considera il nodo G e i suoi figli, il nodo genitore C ha un'etichetta 3 (gioca max). Al livello successivo è min che gioca. Non appena min considera una mossa che lo porta a un figlio di G di etichetta minore o uguale a 3 (ovvero di valore minore o uguale a quello del nodo genitore C dove è max che gioca) può smettere di esplorare gli altri nodi fratelli, poiché se anche qui trovasse una foglia con valore più basso di 3 (e tale strada sceglierebbe min che tende a minimizzare il punteggio) al passo precedente (nodo C) max sceglierebbe la mossa che lo porta verso il nodo F e non verso G, poiché a max interessa invece andare verso il punteggio più alto possibile.

**Se si è lasciato in piedi 11 è perché interessa max? e se è così perché allora è stato tagliata la foglia 22?

NO. Per i nodi B ed E il discorso è simile a quello che lo detto per i nodi C e G.

QUANDO si considera il nodo E ed i suoi figli, il nodo genitore B ha un'etichetta 14 (gioca max). Al livello successivo è min che gioca. Non appena min considera una mossa che lo porta a un figlio di E di etichetta minore o uguale a 14 (ovvero di valore minore o uguale a quello del nodo genitore B dove è max che gioca) può smettere di esplorare gli altri nodi fratelli, poiché se anche qui trovasse una foglia con valore più basso di 14 (e tale strada sceglierebbe min che tende a minimizzare il punteggio) al passo precedente (nodo B) max sceglierebbe la mossa che lo porta verso il nodo D e non verso E, poiché a max interessa invece andare verso il punteggio più alto possibile. Per questo non appena si considera il nodo J e si trova 10 (≤ 14) si tagliano gli altri due rami (che portano a K e L).

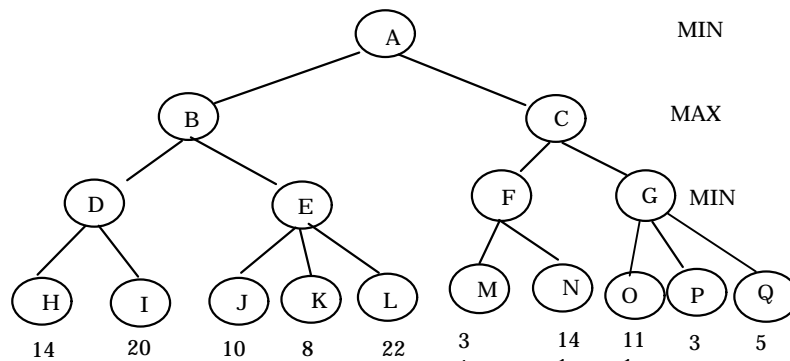
Risposta 2:

Uno stesso albero di ricerca può essere tagliato, secondo l'algoritmo alfa beta in più modi oppure la soluzione è UNICA?

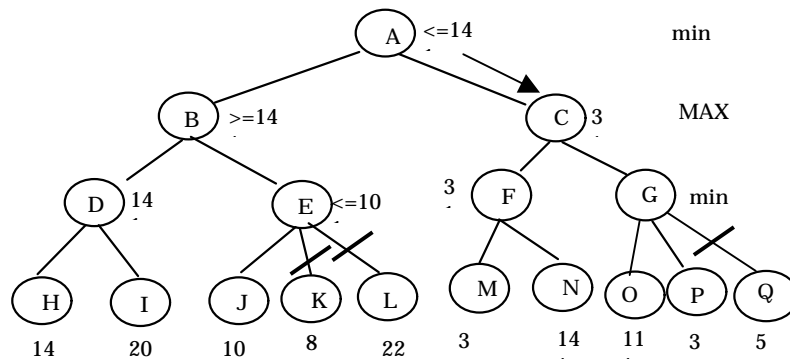
La soluzione è unica.

E cioè, con riferimento per esempio all'esercizio 3 del 11/2/2000:

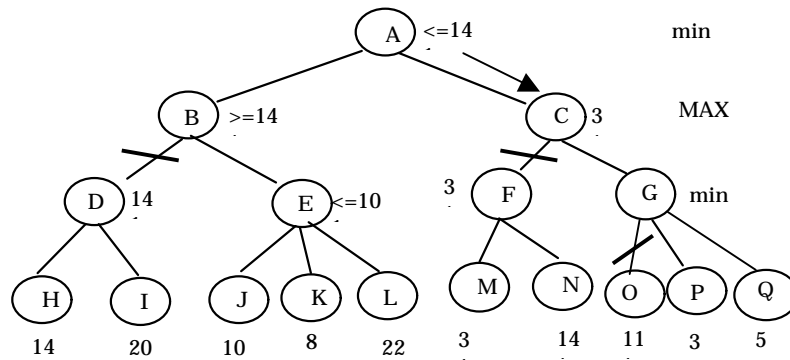
Dato il seguente albero di ricerca per un gioco a due giocatori, si mostri quale mossa selezionerà MIN, secondo l'algoritmo MIN-MAX. Si mostri inoltre come può essere ridotto l'albero applicando i tagli alfa-beta.



Oltre alla soluzione proposta:



Potrebbe essere corretta anche questa?



E se no perché?

Senz'altro non si può eliminare il primo ramo a sinistra, perché è il primo ramo esplorato dall'algoritmo.

Perché nella soluzione da voi proposta nelle foglie 11, 3 e 5 si è scelto di non espandere 5 invece di 11 che è meno interessante per MIN?

Perché l'algoritmo non lo sa; finché non ha esplorato un ramo, non sa quale sia il valore da attribuirgli. L'algoritmo cerca di non esplorare rami che comunque non porteranno a soluzioni "migliori" di quelle già trovate. Nell'albero che consideriamo, l'esplorazione avviene così :

1. Partiamo dal nodo A (non c'è nessuna etichetta)
2. Nodo B: non ci sono etichette
3. Nodo D: idem
4. Nodo H: qui l'albero termina ed abbiamo 14 per la funzione di valutazione. Prima di esplorare questo nodo non sapevamo che avremmo avuto 14, quindi questo dobbiamo esplorarlo per forza.
5. Nodo I: Otteniamo 20, quindi min riporta l'etichetta 14 sul nodo D
6. A questo punto si torna a B e stiamo per esplorare il sottoalbero E; poiché in B è MAX a scegliere, egli sa che non sceglierà mai un ramo che sia peggiore di D. Quindi se, esplorando il sottoalbero E, arriviamo a dimostrare che questo sottoalbero avrà un valore ≤ 14 , possiamo smettere di esplorarlo (MAX comunque sceglierà la strada D). Ecco perché etichettiamo il nodo B con ≥ 14
7. Nodo E: non ci sono etichette
8. Nodo J restituisce 10
9. In E è min a scegliere, quindi cercherà di minimizzare la funzione di valutazione. Quindi, indipendentemente da quel che succede nei nodi K ed L, il valore attribuito ad E sarà ≤ 10 . Ma allora MAX non sceglierà il ramo E, ma preferirà il ramo D, quindi è inutile esplorare i nodi K ed L
10. Nodo C
11. Nodo F
12. Nodo M (3)
13. Nodo N (14)
14. Nel nodo F si riporta 3
15. Nel nodo C sappiamo che MAX otterrà qualcosa di ≥ 3 .
16. Nodo G
17. Nodo O (11)
18. Nodo P (3)
19. A questo punto sappiamo che min sceglierà una strada ≤ 3 , (indipendentemente da quello che succede nel nodo Q), di conseguenza MAX non sceglierà di effettuare la mossa G, ma la F, quindi è inutile esplorare il nodo Q.