

Παιχνίδια

- Τα παιχνίδια με τα οποία θα ασχοληθούμε είναι αυτά που μπορούν να οριστούν αυστηρά, παίζονται από δύο παίκτες και είναι σχετικά απλό να αναπαρασταθεί η κατάσταση τους.
- Γενικά, τα παιχνίδια θα μπορούσαμε να τα κατατάξουμε στην κατηγορία των προβλημάτων όπου η αρχική γνώση δεν είναι αρκετή για την λύση του προβλήματος. Αυτό συμβαίνει διότι δεν ξέρουμε πως σκέφτεται ο αντίπαλος και άρα δεν μπορούμε να γνωρίζουμε ποιο θα είναι το τελικό αποτέλεσμα κάποιας δική μας κίνησης. Ωστόσο, γνωρίζουμε όλα τα πιθανά ενδεχόμενα και έτσι μπορούμε στην γενική περίπτωση να τα εξετάσουμε και να κρίνουμε ποίο από αυτά μας συμφέρει περισσότερο.

Παιχνίδια

- **Γενική Θεώρηση:** Έστω ότι έχουμε τους παίκτες X και Y . Ο κάθε παίκτης, σε κάθε κίνηση που κάνει, προσπαθεί να **μεγιστοποιήσει την πιθανότητά** του να κερδίσει.
- Ο X σε κάθε κίνηση που κάνει προσπαθεί να μεγιστοποιήσει την πιθανότητα του να κερδίσει ενώ ο Y σε κάθε κίνηση του προσπαθεί να ελαχιστοποιήσει την πιθανότητα αυτή (την πιθανότητα να κερδίσει ο X).
- Βολεύει να ονομάζουμε τους παίκτες με τα ονόματα MAX και MIN . MAX είναι ο παίκτης που θέλουμε να νικήσει και ο αλγόριθμος προσπαθεί να διαλέξει τις καλύτερες κινήσεις για αυτόν ενώ MIN είναι ο αντίπαλος.

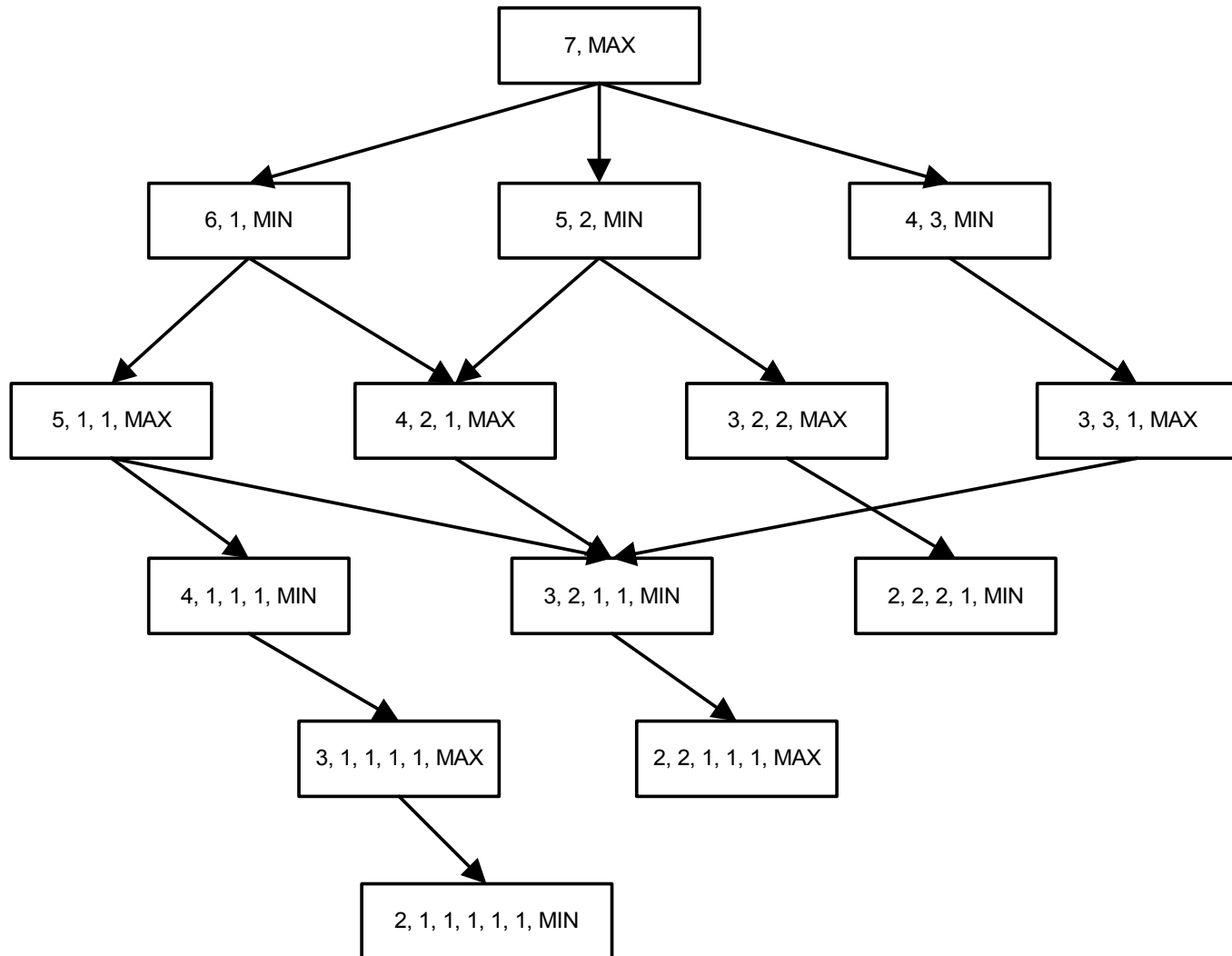
Παίξιμο παιχνιδιών

- Ανεπαρκής η αρχική πληροφορία για την επίλυση του προβλήματος.
- Απαιτείται ειδική διαδικασία αναζήτησης.
- Για μη τετριμμένα παιχνίδια ο χώρος αναζήτησης είναι εξαιρετικά μεγάλος - πρακτικά είναι αδύνατο η εξαντλητική αναζήτηση.
- Σε πολλά παιχνίδια το ζητούμενο δεν είναι μόνο να νικήσει ο παίκτης αλλά να πετύχει και το μεγαλύτερο δυνατό σκορ (βαθμολογία)
- Στην περίπτωση που σημασία έχει μόνο η νίκη ή η ισοπαλία (όπως το σκάκι) μπορούμε να δώσουμε το σκορ 1 για αυτόν που κερδίζει, 0 για αυτόν που κάνει ισοπαλία και -1 για αυτόν που χάνει.

Παράδειγμα - τα 7 δεκάρικα

- Έστω ότι παίζουμε το παιχνίδι με τα 7 δεκαράκια (Grundy's game).
- Σε αυτό το παιχνίδι έχουμε 7 δεκαράκια τοποθετημένα αρχικά σε μία στοίβα. Οι δύο παίκτες παίζουν ο ένας μετά τον άλλο.
- Ο κάθε παίκτης, στην σειρά του πρέπει να διασπάσει μία στοίβα σε δύο άλλες στοίβες με άνισο αριθμό από δεκάρικα.
- Ο παίκτης που δεν μπορεί να το κάνει χάνει.

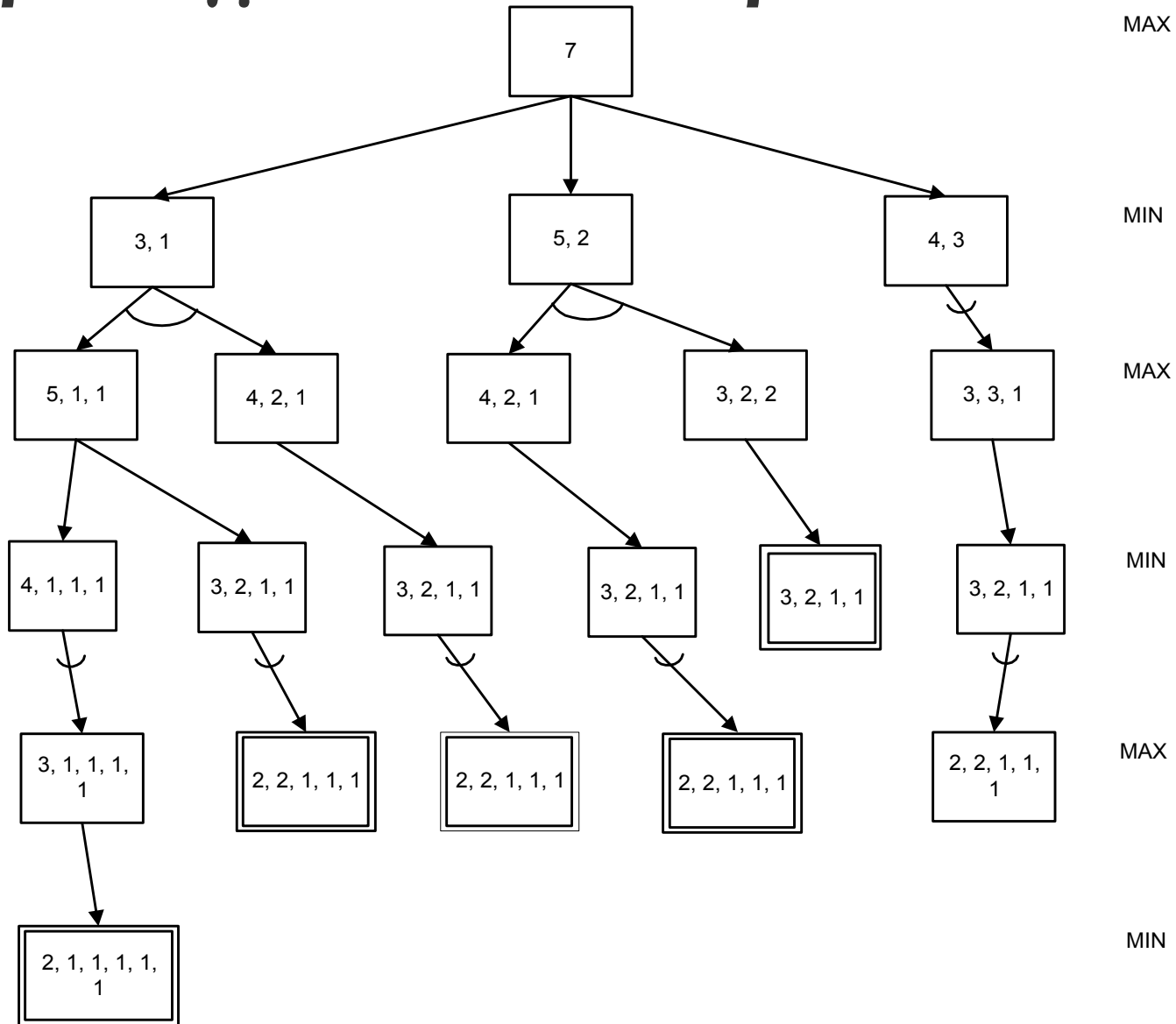
Παράδειγμα - τα 7 δεκάρικα



Παράδειγμα - τα 7 δεκάρικα

- Δεν αρκεί να βρούμε ένα μονοπάτι στο γράφο από την αρχική κατάσταση σε κάποια νικηφόρα κατάσταση. Η ικανότητα του MIN να διαλέγει ποια θα είναι η κίνησή του κάνει την αναζήτηση πιο περίπλοκη.
- Ο πιο κατάλληλος τρόπος για να αναπαραστήσουμε την αναζήτηση είναι να χρησιμοποιήσουμε **δέντρα AND-OR**. Οι απόγονοι των OR κόμβων αντιπροσωπεύουν τις επιλογές κινήσεων που έχει ο παίκτης MAX ενώ οι απόγονοι των AND κόμβων αντιπροσωπεύουν τις επιλογές που έχει ο MIN παίκτης.
- Για να αξίζει κάποια κίνηση του MAX παίκτη (OR κόμβος) πρέπει κάθε επιτρεπτή επόμενη κίνηση του MIN (κάθε παιδί του OR κόμβου) να συμφέρει τον MAX.

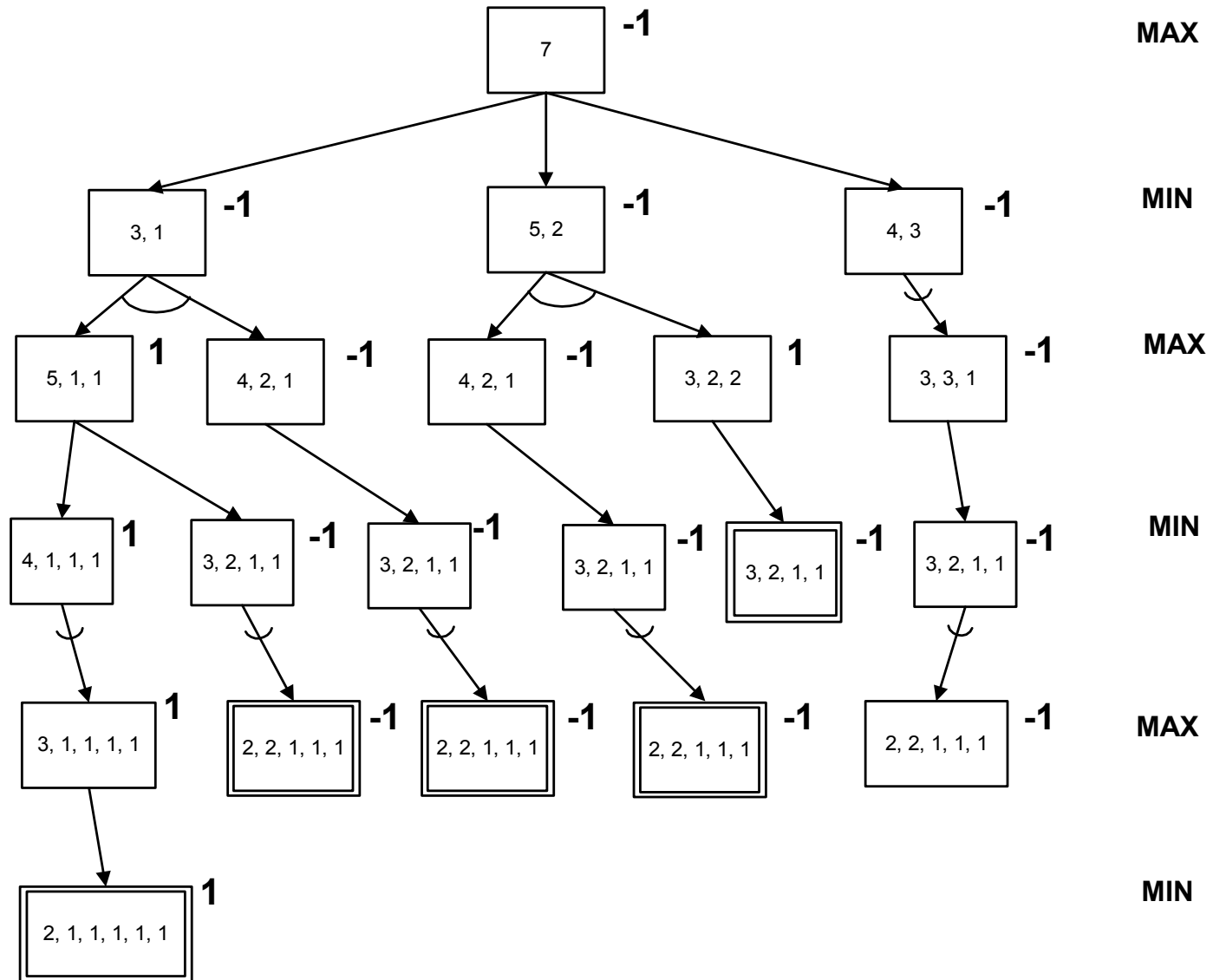
Παράδειγμα - τα 7 δεκάρικα



Η διαδικασία MIN-MAX

- Κατασκευάζουμε το πλήρες δέντρο του παιχνιδιού (το δέντρο AND-OR).
- Βαθμολογούμε τους κόμβους φύλα (τελικές καταστάσεις) σύμφωνα με την βαθμολογία που θα έπαιρνε ο παίκτης MAX αν το παιχνίδι κατέληγε σε εκείνη την κατάσταση.
- Εφαρμόζουμε τα παρακάτω δύο βήματα επαναληπτικά μέχρι όλοι οι κόμβοι του δέντρου να έχουν βαθμολογηθεί:
 - Για κάθε κόμβο MAX (AND κόμβος) τα παιδιά του οποίου έχουν όλα βαθμολογηθεί υπολογίζουμε το μέγιστο αυτών των βαθμών και του αναθέτουμε αυτή την τιμή.
 - Για κάθε κόμβο MIN (OR κόμβος) τα παιδιά του οποίου έχουν όλα βαθμολογηθεί υπολογίζουμε το ελάχιστο αυτών των βαθμών και του αναθέτουμε αυτή την τιμή.

Παράδειγμα MIN-MAX (7 δεκάρικα)



Πρακτικά προβλήματα

- Πρόβλημα : συνήθως τα πλήρη δέντρα είναι **εξαιρετικά μεγάλα** - δεν χωράνε στη μνήμη.
- Λύση : αναπτύσσουμε μερικώς το δέντρο και εκτιμάμε την τιμή για τους κόμβους φύλα.
- Προβλήματα :
 - **πότε σταματάμε** την ανάπτυξη του δέντρου;
 - **πώς υπολογίζουμε την τιμή** στα φύλα;
- Λύσεις :
 - ανάπτυξη μέχρι σταθερό βάθος
 - ευριστική συνάρτηση εκτίμησης αξίας

Η τεχνική Alfa-Beta (α - β)

- Αποφυγή των περιττών πράξεων του MIN-MAX.
- Ο αλγόριθμος Alfa-Beta εκμεταλλεύεται ορισμένες ιδιότητες του αλγορίθμου MIN-MAX και επιτρέπει τον υπολογισμό της επόμενης κίνησης **χωρίς να υπολογίσει την τιμή όλων των κόμβων του δέντρου.**
- Δεν επηρεάζει την απόφαση του αλγορίθμου MIN-MAX αφού κάνει περικοπές μόνο σε κλαδιά τα οποία σίγουρα δεν θα επηρεάσουν την τελική απόφαση.
- Κάνει περικοπή μόνο ασήμαντων κλαδιών :

$\text{MIN}(X_1, X_2, \dots, X_{k-1}, X_k) = \text{MIN}(X_1, X_2, \dots, X_{k-1})$ αν υπάρχει i με $0 < i < k$ και $X_k > X_i$.

$\text{MAX}(X_1, X_2, \dots, X_{k-1}, X_k) = \text{MAX}(X_1, X_2, \dots, X_{k-1})$ αν υπάρχει i με $0 < i < k$ και $X_k < X_i$.

Η τεχνική Alfa-Beta (α - β)

- Έστω, για παράδειγμα ότι πρέπει να υπολογιστεί η τιμή ενός MIN κόμβου. Έχουμε υπολογίσει την τιμή του πρώτου παιδιού του και για το επόμενο παιδί του ξέρουμε (με κάποιον τρόπο) ότι η τιμή του θα είναι μεγαλύτερη από την τιμή του πρώτου παιδιού. **Μπορούμε με σιγουριά** να πούμε ότι η ακριβής τιμή του δεύτερου παιδιού δεν θα αλλάξει σε τίποτα την τελική τιμή του κόμβου. **Δεν αλλοιώνει** το αποτέλεσμα του MIN-MAX.
- Μπορεί να χρησιμοποιηθεί και κατά την διάρκεια ανάπτυξης του δέντρου.
- Έχει σημασία η σειρά εξέτασης των παιδιών ενός κόμβου.
- Σταματάμε την εξέταση ενός κόμβου όταν ισχύει $\text{Alfa} > \text{Beta}$.

Ο αλγόριθμος Alfa-Beta

1. Θέσε α = ένας πολύ μεγάλος αρνητικός αριθμός.
2. Θέσε β = ένας πολύ μεγάλος θετικός αριθμός.
3. Θέσε ρ = η θέση από την οποία πρέπει να παίξει ο παίκτης MAX.
4. Η minmax τιμή του κόμβου ρ = $TIMH_MAX(\rho, \alpha, \beta)$

$TIMH_MAX(\rho, \alpha, \beta)$

1. Αν Είναι_Φύλλο(ρ) τότε επέστρεψε Αξία_Κόμβου(ρ).
2. Διαφορετικά εκτέλεσε τα παρακάτω βήματα για κάθε παιδί π του ρ .
 - 2.1. Θέσε $\alpha = MAX(\alpha, TIMH_MIN(\pi, \alpha, \beta))$
 - 2.2. Αν $\alpha \geq \beta$ επέστρεψε β .
3. Επέστρεψε α .

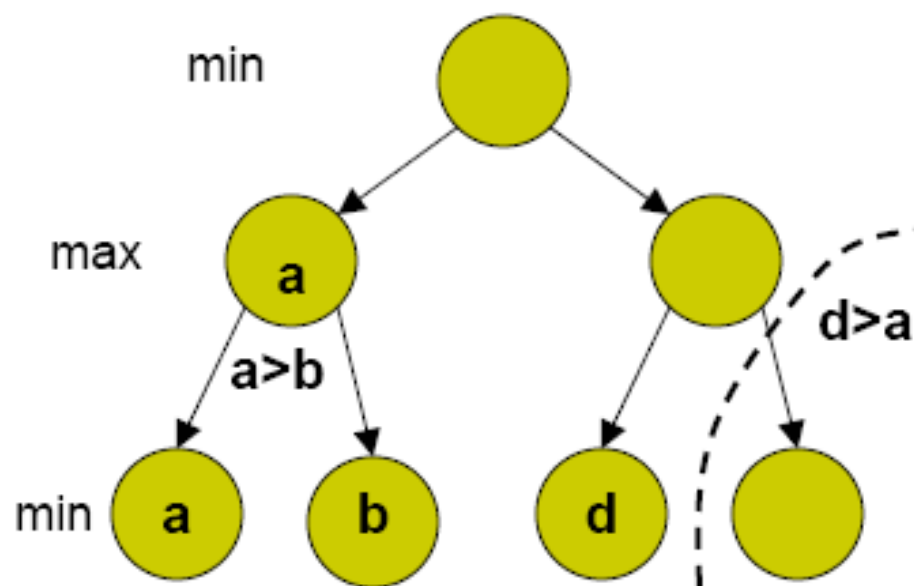
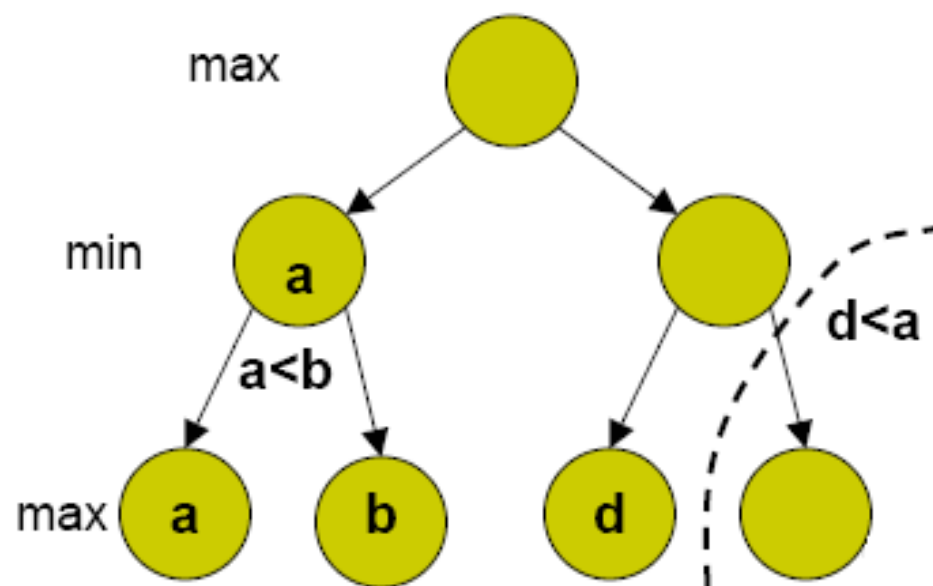
$TIMH_MIN(\rho, \alpha, \beta)$

1. Αν Είναι_Φύλλο(ρ) τότε επέστρεψε Αξία_Κόμβου(ρ).
2. Διαφορετικά εκτέλεσε τα παρακάτω βήματα για κάθε παιδί π του ρ .
 - 2.1. Θέσε $\beta = MIN(\beta, TIMH_MAX(\pi, \alpha, \beta))$
 - 2.2. Αν $\alpha \geq \beta$ επέστρεψε α .
3. Επέστρεψε β .

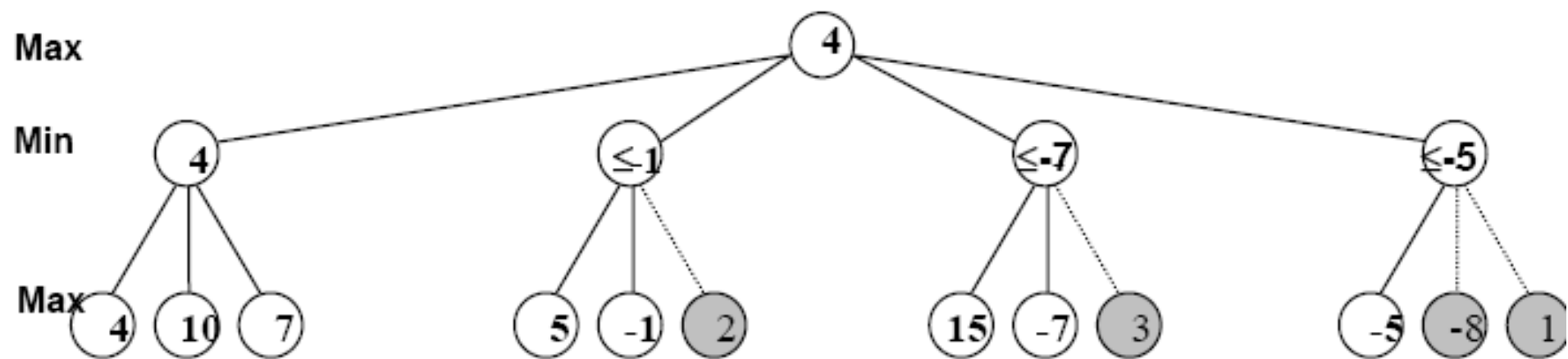
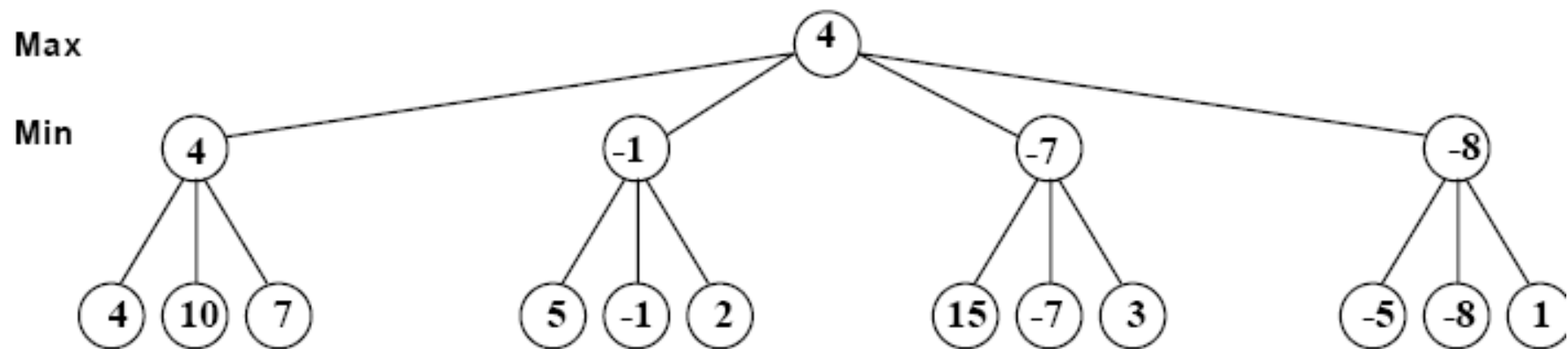
Τεχνική κλαδέματος Alpha-Beta

(Alpha-Beta pruning)

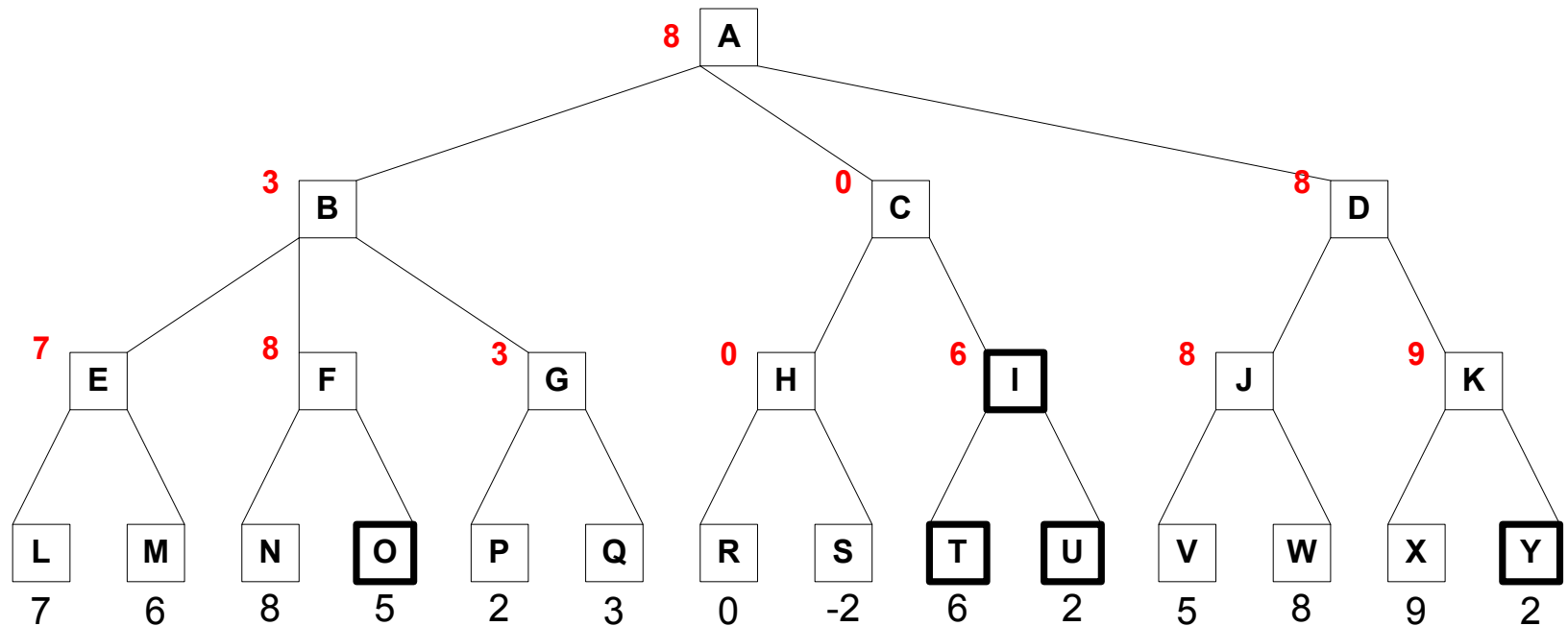
- Η τεχνική κλαδέματος Alpha-Beta εφαρμόζεται στον MiniMax, με σκοπό την εξέταση (βαθμολόγηση) λιγότερων καταστάσεων στον ορίζοντα της αναζήτησης.
 - Οι καταστάσεις στον ορίζοντα αναζήτησης δεν βαθμολογούνται όλες μαζί, αλλά δίνεται προτεραιότητα στο γρήγορο "ανέβασμα" των τιμών.



Παράδειγμα Alpha-Beta pruning



Παράδειγμα



- Ο Alfa-Beta **δεν θα εξετάσει** τους κόμβους : O, I, T, U, Y.
- Επιπλέον, για τους κόμβους F, C, K **δεν θα υπολογιστεί** η ακριβής MINMAX τιμή τους.