



ΜΕΡΟΣ Α

Κεφάλαιο 5



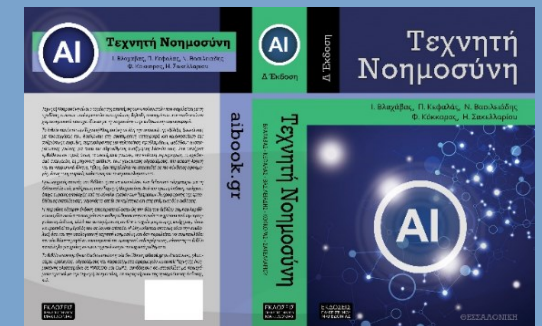
Αλγόριθμοι Αναζήτησης σε Παίγνια Δύο Αντιπάλων

Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου.

Τεχνητή Νοημοσύνη - Δ' Έκδοση, ISBN: 978-618-5196-44-8

Έκδοση/Διάθεση: Εκδόσεις Πανεπιστημίου Μακεδονίας, 2020

Κωδικός Βιβλίου στον Εύδοξο: 94700120





Αλγόριθμοι Αναζήτησης σε Παίγνια Δύο Αντιπάλων¹

- ❖ Σε πολλά προβλήματα της ΤΝ, η εξέλιξη των καταστάσεων εξαρτάται από δύο διαφορετικά σύνολα τελεστών μετάβασης που εφαρμόζονται εναλλάξ από δύο ενεργά συστήματα (πράκτορες-agents).
- ❖ Αυτά τα προβλήματα αναφέρονται και ως ανταγωνιστικά παίγνια ή παίγνια δύο αντιπάλων (*adversary* ή *two-person games*)
 - ❑ Επιλύονται με ειδικούς αλγορίθμους αναζήτησης, βασικότεροι των οποίων είναι ο *minimax* και ο *alpha-beta*.
- ❖ Ο όρος "**παίγνιο**" αφορά την περιγραφή του τρόπου με τον οποίο παίζεται το παιχνίδι και περιλαμβάνει:
 - ❑ τα αντικείμενα που υπάρχουν (για παράδειγμα, τα πούλια, το ταμπλώ, κτλ.) καθώς και
 - ❑ το σύνολο των κανόνων (για παράδειγμα, των επιτρεπτών κινήσεων, των συνθηκών νίκης, κτλ.) που το διέπουν.
 - ❑ Γνωστά παραδείγματα παιγνίων είναι το σκάκι, η τρίλιζα, το τάβλι, κτλ. ([Tic-tac-toe](#), [Chess](#), [Go](#), etc.)
- ❖ Αντίθετα, με τον όρο "**παιχνίδι**" χαρακτηρίζεται μία συγκεκριμένη παρτίδα του παιγνίου.

¹<https://en.wikipedia.org/wiki/Minimax>

https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning



Ορισμός του Προβλήματος

- ❖ Σε ένα παίγνιο δύο ατόμων το πρόβλημα ορίζεται ως εξής:
 - ☐ Μια κατάσταση παριστάνει τη διάταξη των πιονιών σε κάποια χρονική στιγμή.
 - ☐ Ο χώρος καταστάσεων αποτελείται από όλες αυτές τις πιθανές επιτρεπτές καταστάσεις.
 - ☐ Οι τελεστές μετάβασης είναι οι επιτρεπτές κινήσεις που καθορίζονται από τους κανόνες του παιχνιδιού.
 - ✓ Οι κανόνες του παιχνιδιού παίζουν και το ρόλο των προϋποθέσεων αυτών των τελεστών.
 - ☐ Οι τελικές καταστάσεις του παιχνιδιού δεν είναι γνωστές αλλά έχουν γνωστά χαρακτηριστικά (π.χ. ματ στο σκάκι).
- ❖ Έστω ότι κάποιος συμμετέχει σε ένα παιχνίδι και είναι η σειρά του να κάνει μία κίνηση.
 - ☐ Αν θέλει να κερδίσει, θα επιδιώξει να κάνει την καλύτερη κίνηση για αυτόν, η οποία αντίστοιχα θα είναι και η χειρότερη για τον αντίπαλο.
 - ☐ Ο τρόπος που πιθανά σκέφτεται κάποιος για να παίξει είναι ο εξής: "Αν κάνω αυτή την κίνηση, τότε ο αντίπαλος θα κάνει εκείνη, αν όμως κάνω την άλλη κίνηση, τότε θα κάνει αυτήν, ..." κ.ο.κ.
 - ☐ Ο συλλογισμός αυτός αντιστοιχεί στη δημιουργία ενός **δένδρου** με όλες τις εναλλακτικές κινήσεις που πηγάζουν από την παρούσα κατάσταση.
 - ☐ Το δένδρο αυτό ονομάζεται **δένδρο του παιχνιδιού (game tree)**.



Δένδρο Παιχνιδιού (game tree) (1/2)

- ❖ Το χαρακτηριστικό του είναι ότι οι κινήσεις δύο διαδοχικών επιπέδων ανήκουν σε διαφορετικό παίκτη, γιατί οι παίκτες παίζουν εναλλάξ.
 - ❑ Το δένδρο αυτό αναπτύσσεται νοερά από τον παίκτη του οποίου είναι η σειρά να παίξει.
 - ❑ Αυτή η διαδικασία συνεχίζεται μέχρι κάποιο βάθος (συνήθως μικρό), σταματά και αρχίζει η αξιολόγηση των καταστάσεων με βάση κάποια κριτήρια.
 - ❑ Με βάση την αξιολόγηση, ο παίκτης αποφασίζει ποια από τις εναλλακτικές κινήσεις θα τον οδηγήσει σταδιακά στην πιο ευνοϊκή για αυτόν εξέλιξη.
- ❖ Έστω ότι ένας από τους δύο αντιπάλους είναι ο υπολογιστής.
 - ❑ Το πρόγραμμα αναζήτησης και ο άνθρωπος πρέπει να εξετάσουν και να θυμούνται όλες τις πιθανές κινήσεις που παράγονται από μία (αρχική) κατάσταση.
 - ❑ Ένα τέτοιο πρόγραμμα μπορεί να προβλέψει την έκβαση του παιχνιδιού μετά από 10 κινήσεις (βάθος 10 του δένδρου παιχνιδιού), ενώ ο άνθρωπος σταματά συνήθως μετά την πρόβλεψη 2 ή 3 κινήσεων.
- ❖ Παίγνια στα οποία όλες οι πληροφορίες που αφορούν και τους δύο παίκτες είναι γνωστές ανά πάσα στιγμή ονομάζονται **παίγνια απόλυτης πληροφορίας (full information games)**.
 - ❑ Η διαχείριση της πληροφορίας αυτής και η αξιολόγηση των καταστάσεων για κάθε παίκτη γίνεται με καθαρά υποκειμενικά κριτήρια και βασίζεται στην εμπειρία και γνώση που έχει για το πρόβλημα.
 - ❑ Αυτό παραπέμπει στη γνωστή έννοια των ευρετικών μηχανισμών.
 - ❑ Για το λόγο αυτό η αναζήτηση σε δένδρα παιχνιδιών κατατάσσεται στη γενικότερη κατηγορία της ευρετικής αναζήτησης.



Δένδρο Παιχνιδιού (game tree) (2/2)

- ❖ Κατά την περασμένη δεκαετία κατασκευάστηκαν συστήματα (συνδυασμός λογισμικού και υλικού), όπως το DEEP BLUE και το DEEP FRITZ, αλλά και κοινά εμπορικά προγράμματα, όπως το JUNIOR και FRITZ, που μπορούν κάποιες φορές να νικήσουν τον παγκόσμιο πρωταθλητή στο σκάκι.
- ❖ Μία αντίστοιχη προσπάθεια της GOOGLE στέφτηκε με επιτυχία όταν το ALPHAGO κέρδισε τον παγκόσμιο πρωταθλητή στο παιχνίδι Go, ένα στρατηγικό κινέζικο παιχνίδι που παίζεται σε ταμπλώ 19x19.
- ❖ Παραδόξως, οι δυνατότητες που έχουν αυτά τα συστήματα δεν κάνουν κάποιο πρόγραμμα κυρίαρχο οποιουδήποτε παιχνιδιού.
- ❖ Η ανωτερότητα τους έγκειται στους εξής παράγοντες:
 - ❑ Οι μαιτρ του σκάκι έχουν τη δυνατότητα να **επιλέγουν με κάποιον ευρετικό τρόπο** τις εναλλακτικές κινήσεις, αλλά δε μπορούν να μετατρέψουν αυτόν τον τρόπο σε αντίστοιχη ευρετική τιμή και να υλοποιηθεί σαν πρόγραμμα.
 - ❑ Οι άνθρωποι διαθέτουν κάποια διαίσθηση για την κατάληξη του παιχνιδιού αν επιλεγεί κάποια από τις εναλλακτικές κινήσεις η οποία επίσης δε μπορεί να αποδοθεί αριθμητικά ώστε να υπάρξει ευρετικό κλάδεμα του χώρου αναζήτησης.
 - ❑ Τέλος, με την εμπειρία που αποκτούν, μπορούν να σκέφτονται πολλές φορές εντελώς μηχανικά, ιδίως στα πρώτα και τελευταία στάδια του παιχνιδιού.



Προγράμματα Σκάκι

- ❖ Το κύριο μέλημα τους είναι να αναζητήσουν το δένδρο του παιχνιδιού σε όσο το δυνατόν μεγαλύτερο βάθος.
 - ❑ Αν υπήρχε η δυνατότητα να φτάσουν μέχρι τις τελικές καταστάσεις του παιχνιδιού τότε τα προγράμματα θα ήταν ανίκητα.
 - ❑ Ο λόγος που κάτι τέτοιο είναι αδύνατο πηγάζει από τους χρονικούς περιορισμούς μιας παρτίδας και από το πρόβλημα της συνδυαστικής έκρηξης.
- ❖ Έστω ότι κατά μέσο όρο για κάθε κατάσταση στο σκάκι παράγονται άλλες 10 καταστάσεις (δυνατές κινήσεις) και ότι μία παρτίδα τελειώνει σε 40 κινήσεις (20 για κάθε παίκτη). Τότε οι πιθανές τελικές καταστάσεις θα ήταν της τάξης του 10^{40} .
 - ❑ Αν υποθέσουμε ότι ένας υπέρ-υπολογιστής θέλει 1 nanosecond ($=10^{-9}$ sec) για να εξετάσει κάθε μία από αυτές, τότε χρειάζεται συνολικά $10^{40} \times 10^{-9} = 10^{31}$ sec για να τις εξετάσει όλες.
 - ❑ Με δεδομένο ότι ένα έτος έχει 3×10^7 sec περίπου, ο υπέρ-υπολογιστής αυτός θα χρειαζόταν 3×10^{24} χρόνια για να διαλέξει την επόμενη κίνησή του.
 - ❑ Αυτό είναι περίπου το 1/4 της ηλικίας του σύμπαντος!
- ❖ Αφού λοιπόν δε γίνεται να περιμένουμε τόσο, οι αλγόριθμοι αναζήτησης εξετάζουν το δένδρο του παιχνιδιού μέχρι ένα ορισμένο βάθος, όσο επιτρέπει ο χρόνος μας.





Ο Αλγόριθμος Minimax (1/2)

- ❖ Δεδομένης μίας κατάστασης του παιχνιδιού, ο αλγόριθμος αναζήτησης μεγίστου-ελαχίστου (**Minimax**) καλείται να αποφασίσει ποια θα είναι η επόμενη κίνησή του έναντι του αντιπάλου.
- ❖ Η εξαντλητική αναζήτηση των δένδρων παιχνιδιού είναι ανέφικτη.
- ❖ Το ζητούμενο είναι:
 - ☐ Να χτιστεί το δένδρο μέχρι κάποιο βάθος
 - ☐ Να βρεθεί η καλύτερη κίνηση από την παρούσα κατάσταση.
- ❖ Το μέτρο της υπεροχής του ενός ή του άλλου αντιπάλου δίνεται από μία ευρετική συνάρτηση που καλείται **συνάρτηση αξιολόγησης** (*evaluation function*) και η οποία εφαρμόζεται στα φύλλα του δένδρου του παιχνιδιού.
- ❖ Ο ένας παίκτης (πρόγραμμα) ονομάζεται **max** και ο άλλος (άλλο πρόγραμμα ή άνθρωπος) ονομάζεται **min**.
- ❖ Οι κόμβοι του δένδρου διακρίνονται σε:
 - ☐ Αυτούς που σειρά έχει να παίξει ο max
 - ☐ Αυτούς που σειρά έχει να παίξει ο min





Αλγόριθμος

Αφού δημιουργηθεί το δένδρο παιχνιδιού μέχρι ένα ορισμένο βάθος:

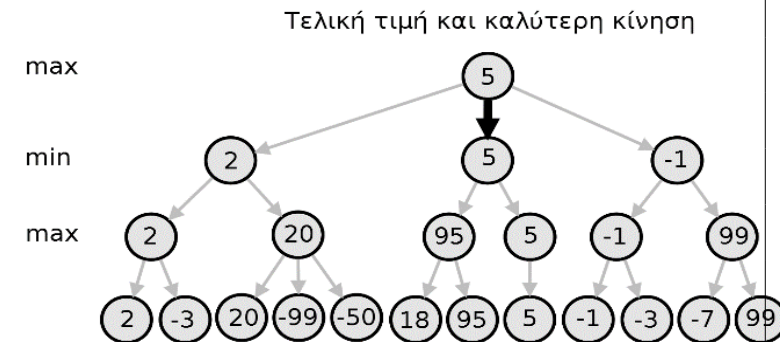
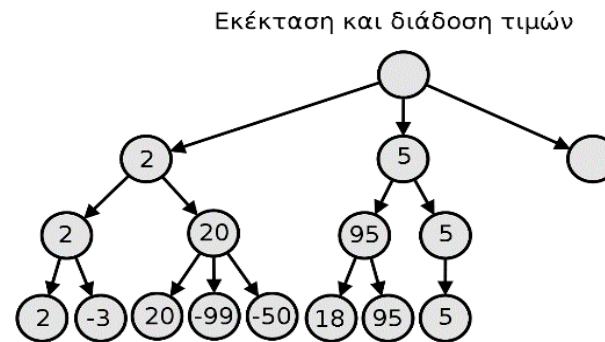
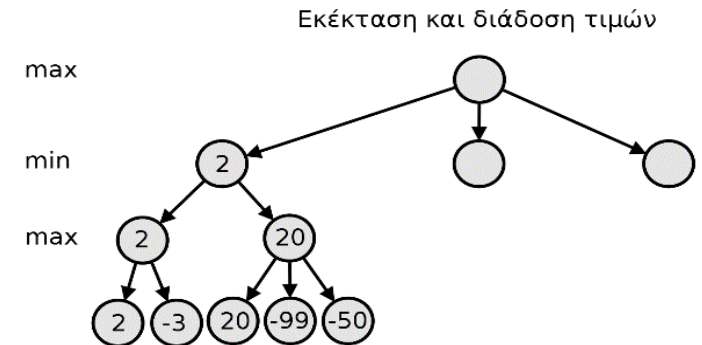
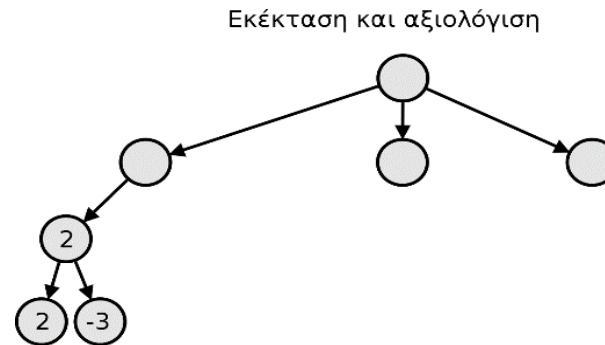
1. Εφάρμοσε τη συνάρτηση αξιολόγησης σε όλους τους κόμβους-φύλλα του δένδρου.
2. Έως ότου η ρίζα του δένδρου αποκτήσει τιμή, επανέλαβε:
3. Αρχίζοντας από τα φύλλα του δένδρου και προχωρώντας προς τη ρίζα, μετέφερε τις τιμές προς τους ενδιαμέσους κόμβους του δένδρου ως εξής:
 - i. Η τιμή κάθε κόμβου **Max** είναι η μέγιστη (**maximum**) των τιμών των κόμβων-παιδιών του.
 - ii. Η τιμή κάθε κόμβου **Min** είναι η ελάχιστη (**minimum**) των τιμών των κόμβων-παιδιών του.
4. Καλύτερη κίνηση είναι η κίνηση που οδηγεί στον κόμβο που έδωσε την πιο συμφέρουσα στη ρίζα τιμή (μέγιστη για το **Max**, ελάχιστη για το **Min**).

❖ Ο αλγόριθμος εγγυάται την πιο συμφέρουσα εξέλιξη μετά από κάποιες κινήσεις, έστω και αν ο αντίπαλος διαλέγει τις καλύτερες για αυτόν κινήσεις.



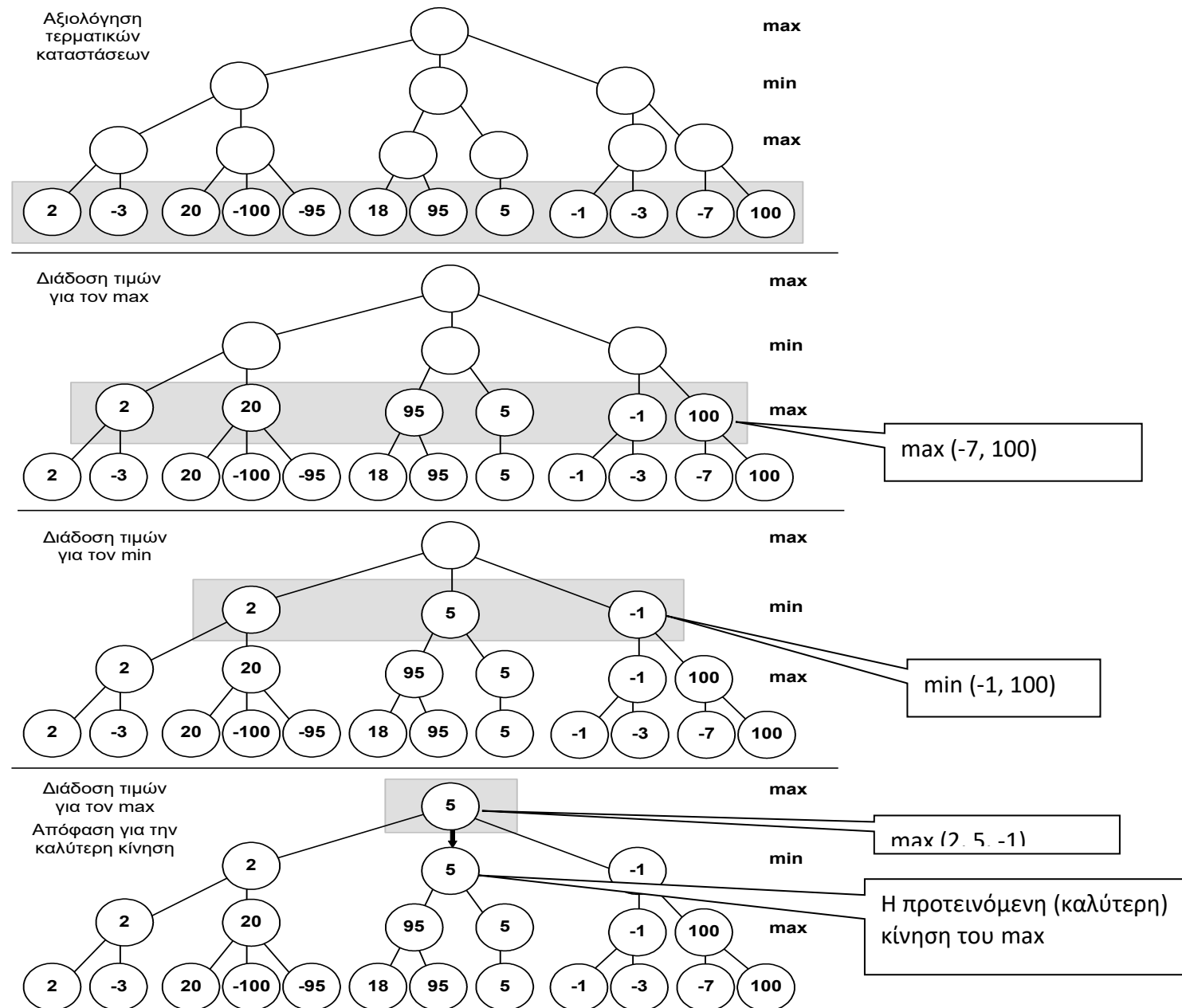
Ο Αλγόριθμος Minimax (2/2)

- ❖ Κατά σύμβαση, ο παίκτης που βρίσκεται στη ρίζα θεωρείται πως είναι ο max.
- ❖ Οι καταστάσεις-φύλλα του δένδρου ονομάζονται και τερματικές καταστάσεις, οι οποίες όμως δεν είναι απαραίτητα τελικές καταστάσεις, αλλά αποτελούν το όριο της αναζήτησης.
- ❖ Οι τιμές των τερματικών καταστάσεων υπολογίζονται από τη συνάρτηση αξιολόγησης ενώ οι άλλες προκύπτουν από τη διάδοση αυτών.
- ❖ Η δημιουργία του δένδρου, η αξιολόγηση και η διάδοση των τιμών γίνεται προοδευτικά, δηλαδή κατά τη διάρκεια της αναζήτησης από αριστερά προς τα δεξιά.





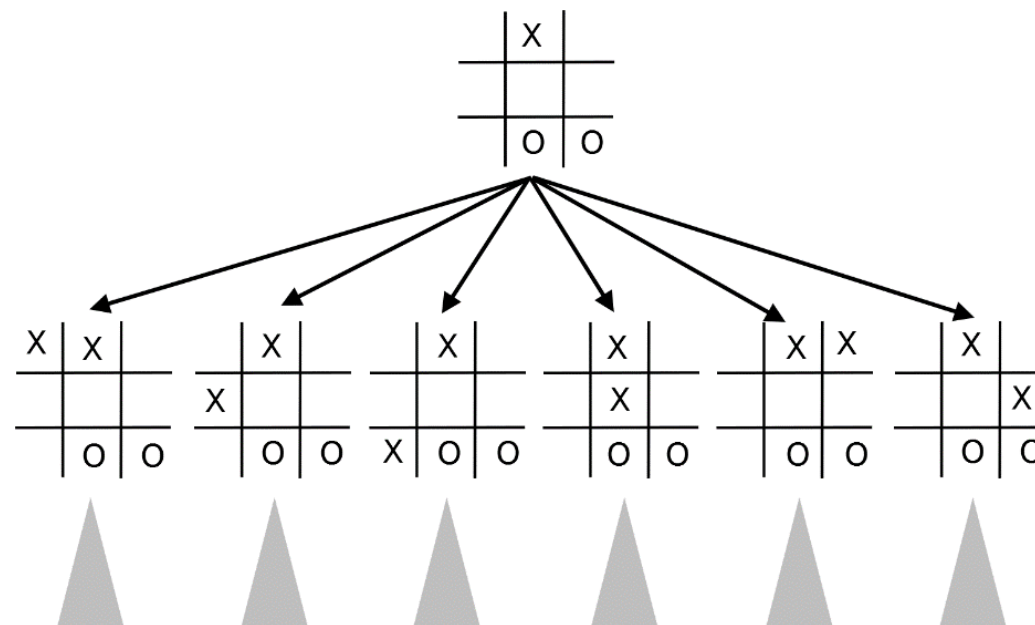
Ο Αλγόριθμος Minimax -Παράδειγμα





Ο Αλγόριθμος Minimax στην Τρίλιζα (1/2)

- ❖ Το παίγνιο τρίλιζα είναι ένα πρόβλημα με μικρό χώρο αναζήτησης, της τάξης των $9!$ καταστάσεων (362.880)
 - ❑ Όλες οι καταστάσεις μπορεί κάλλιστα να κωδικοποιηθούν σε κάποιο πρόγραμμα το οποίο σαν αποτέλεσμα δε θα χάνει ποτέ.
 - ❑ Αποτελεί καλό παράδειγμα εφαρμογής του αλγορίθμου Minimax.
- ❖ Έστω ότι ο παίκτης max είναι ένα πρόγραμμα που σημειώνει: **X** και αντιμετωπίζει έναν άνθρωπο ή ένα άλλο πρόγραμμα, τον min (που σημειώνει: **O**).
 - ❑ Έστω ότι ο max αποφασίζει να δει μέχρι βάθος τριών κινήσεων.
 - ❑ Στο βάθος αυτό εφαρμόζει τη συνάρτηση αξιολόγησης για όλες τις καταστάσεις που έχουν προκύψει.





Ο Αλγόριθμος Minimax στην Τρίλιζα (2/2)

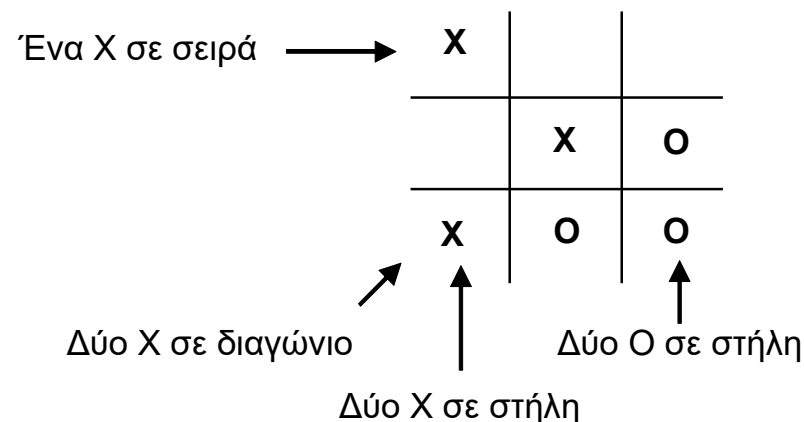
❖ Μία συνάρτηση αξιολόγησης στην τρίλιζα θα μπορούσε να είναι η:

$$3 \cdot X_2 + X_1 - (3 \cdot O_2 + O_1)$$

- ✓ X_2 ο αριθμός γραμμών, στηλών ή διαγώνιων με δύο X και χωρίς κανένα O
- ✓ X_1 ο αριθμός γραμμών, στηλών ή διαγώνιων με ένα X και χωρίς κανένα O
- ✓ O_2 ο αριθμός γραμμών, στηλών ή διαγώνιων με δύο O και χωρίς κανένα X
- ✓ O_1 ο αριθμός γραμμών, στηλών ή διαγώνιων με ένα O και χωρίς κανένα X

❖ Προφανώς, θετικές τιμές επιστρέφουν οι καταστάσεις που ωφελούν αυτόν που σημειώνει X, ενώ αρνητικές τιμές αυτές που ωφελούν αυτόν που σημειώνει O.

- Για παράδειγμα, στο επόμενο Σχήμα απεικονίζεται μία κατάσταση της οποίας η συνάρτηση αξιολόγησης επιστρέφει 4.
- Άρα αυτός που σημειώνει το X έχει εμφανές πλεονέκτημα.



$$3 \cdot X_2 + X_1 - (3 \cdot O_2 + O_1) = 3 \cdot 2 + 1 - (3 \cdot 1 + 0) = 4$$

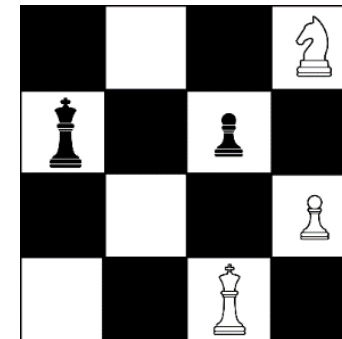


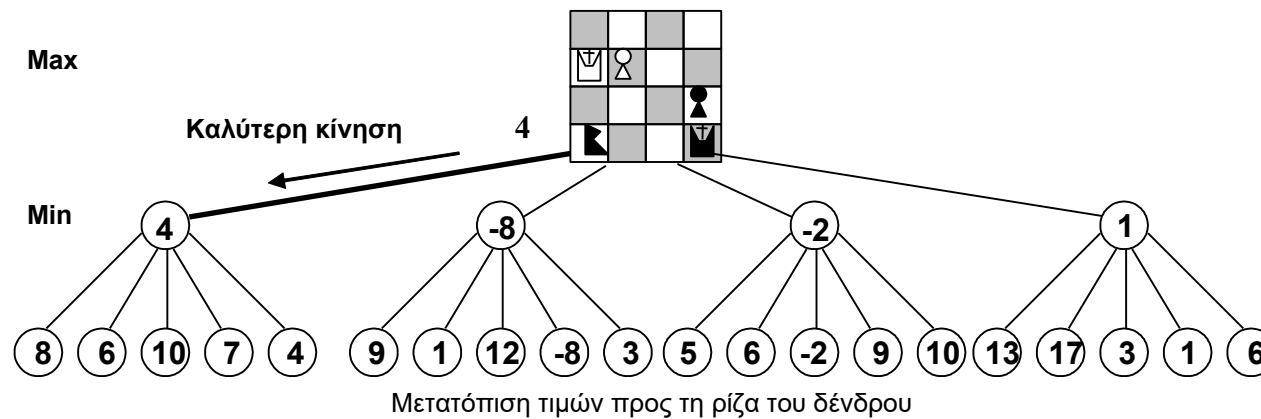
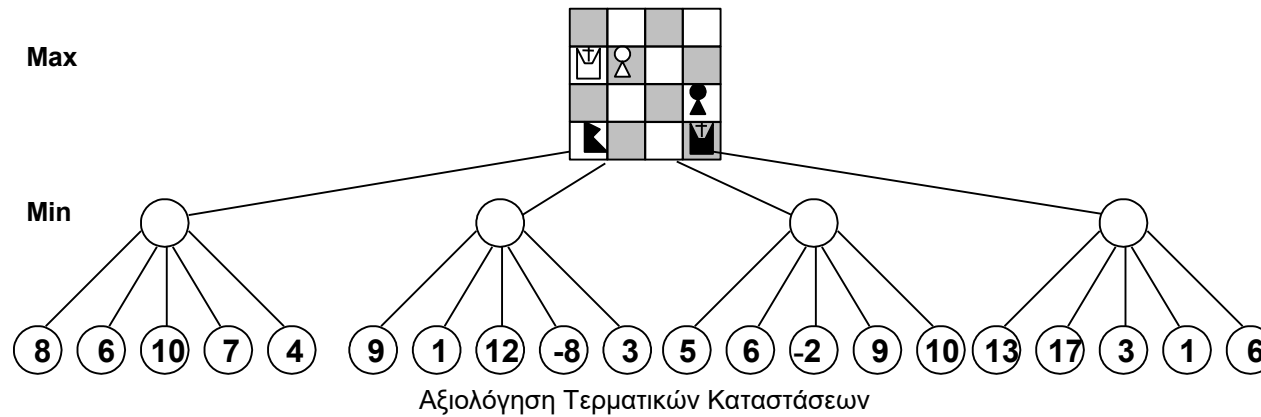
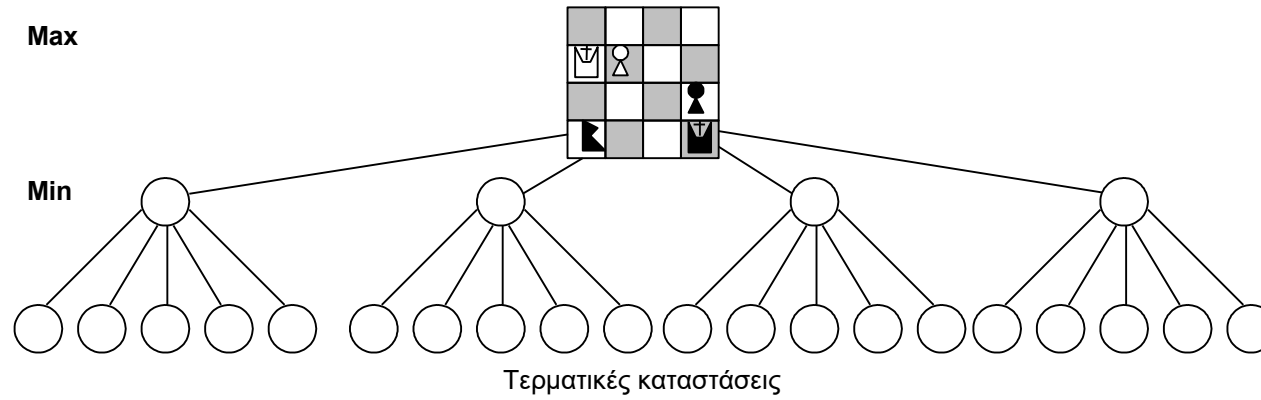
Εφαρμογή αλγορίθμου Minimax στο σκάκι

- ❖ Το κύριο μέλημα των προγραμμάτων σκάκι είναι να αναζητήσουν το δένδρο του παιχνιδιού σε όσο το δυνατόν μεγαλύτερο βάθος.
 - ❑ Αν υπήρχε η δυνατότητα να φτάσουν μέχρι τις τελικές καταστάσεις του παιχνιδιού, τότε τα προγράμματα θα ήταν ανίκητα.

Μία συνάρτηση αξιολόγησης στο σκάκι

- ❑ *Υπεροχή κομματιών*: Κάθε κομμάτι έχει κάποια αξία, π.χ. Βασιλιάς=10, Άλογο=5, Πιόνι=1 κλπ. Η αξία όλων των κομματιών κάθε χρώματος προστίθεται.
- ❑ *Υπεροχή θέσης*: Κάθε κομμάτι που βρίσκεται στα 4 κεντρικά τετράγωνα παίρνει επιπλέον 2 πόντους.
- ❑ *Απειλές*: Για κάθε απειλή ένας παίκτης παίρνει 4 επιπλέον πόντους αν είναι η σειρά του να παίξει ή 3 επιπλέον πόντους αν είναι η σειρά του αντιπάλου., εκτός αν απειλεί το βασιλιά του άλλου παίκτη, οπότε παίρνει 20 πόντους.
- ❑ Αν για παράδειγμα μια κατάσταση σε ένα παιχνίδι σκάκι είναι αυτή του σχήματος:
- ❑ Υποθέτοντας ότι είναι σειρά των μαύρων να παίξουν, η συνάρτηση αξιολόγησης στην εικονιζόμενη κατάσταση, θα έδινε:
 - ✓ Μαύρα: $10+1$ (λόγω α) $+ 2$ (λόγω β) $+ 4$ (λόγω γ) $= 17$
 - ✓ Λευκά: $10+5+1$ (λόγω α) $+ 0$ (λόγω β) $+ 3$ (λόγω γ) $= 19$
 - ✓ οπότε, τα λευκά έχουν υπεροχή $19-17=2$, έναντι των μαύρων.



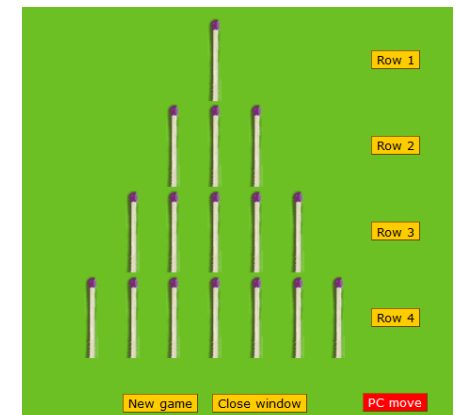
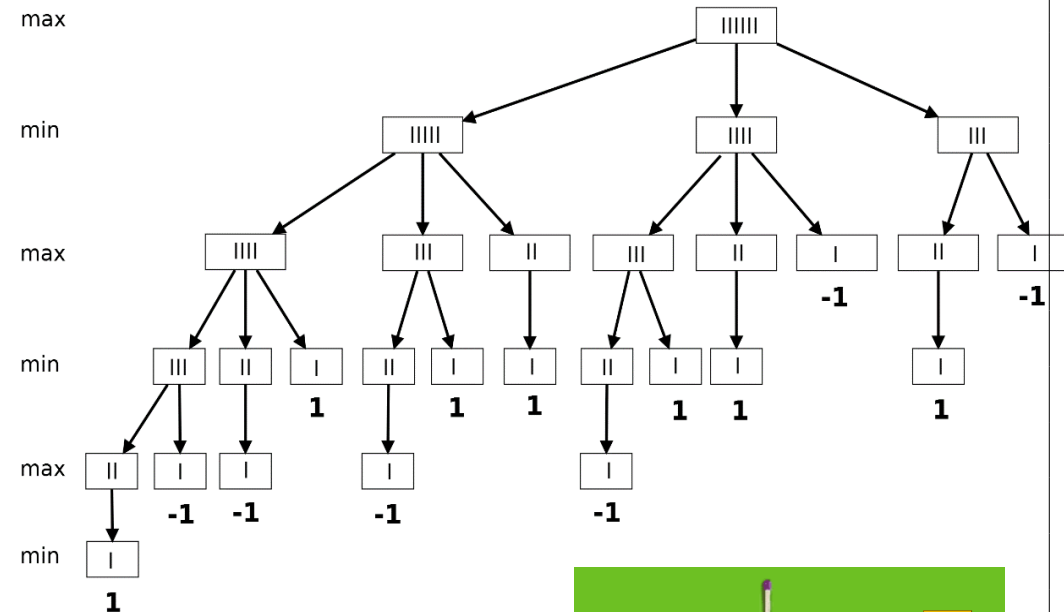




Το παιχνίδι NIM

- ❖ Υπάρχει ένας αριθμός από ξυλάκια πάνω στο τραπέζι, έστω N με $N > 3$.
 - ❑ Κάθε παίκτης παίζει με τη σειρά και μπορεί να πάρει 1 ή 2 ή 3 ξυλάκια, έτσι ώστε στο τραπέζι μένουν $N-1$ ή $N-2$ ή $N-3$ αντίστοιχα.
 - ❑ Νικητής είναι αυτός που στο τέλος αφήνει μόνο 1 για τον αντίπαλο
 - ❖ Για $N=6$ το πλήρες δένδρο παιχνίδια απεικονίζεται στο ακόλουθο Σχήμα.
- ❑ Ο Max παίζει πρώτος.
 - ❑ Η συνάρτηση αξιολόγησης στις τερματικές καταστάσεις επιστρέφει 1 όταν ο Max έχει αφήσει ένα ξυλάκι στο τραπέζι για τον Min (οπότε κερδίζει ο Max) αλλιώς -1 (οπότε χάνει ο Max).

Online Game: https://education.jlab.org/nim/s_index.html

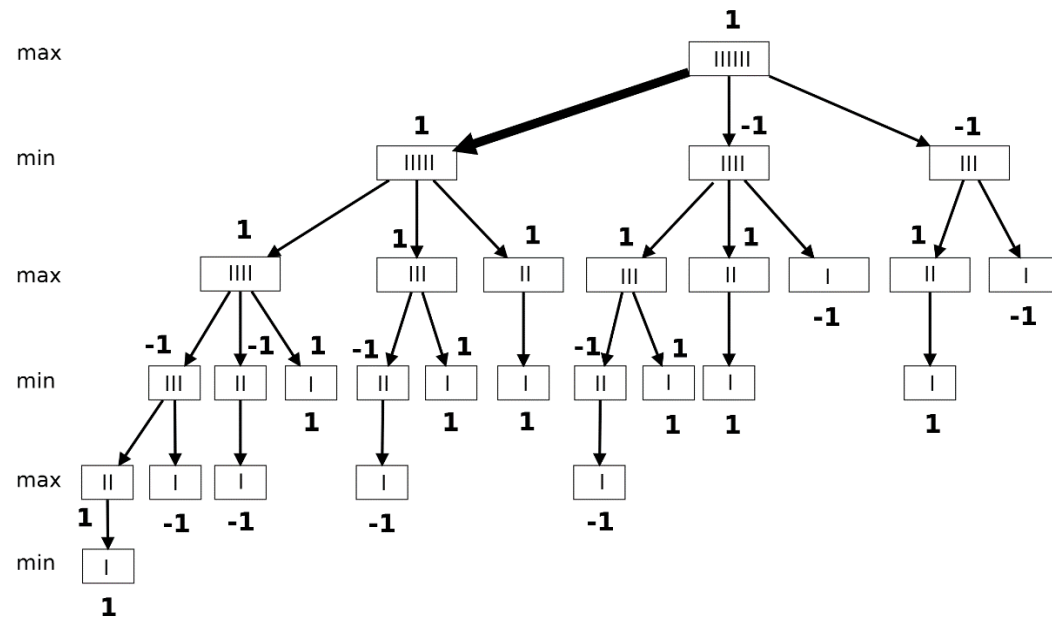


Παραλλαγή: Τα ξυλάκια είναι οργανωμένα σε μορφή δένδρου και ο παίκτης είναι υποχρεωμένος να παίρνει ξυλάκια από 1 σειρά μόνο

Online Game: https://www.archimedes-lab.org/game_nim/play_nim_game.html



- ❖ Εφαρμόζοντας τον αλγόριθμο Minimax, οι τιμές 1 και -1 διαδίδονται προς τη ρίζα του δένδρου στην οποία η τελική τιμή που φτάνει είναι 1 και προέρχεται από το αριστερότερο κλαδί, δηλαδή την κίνηση του Max να πάρει 1 ξυλάκι από τα 6

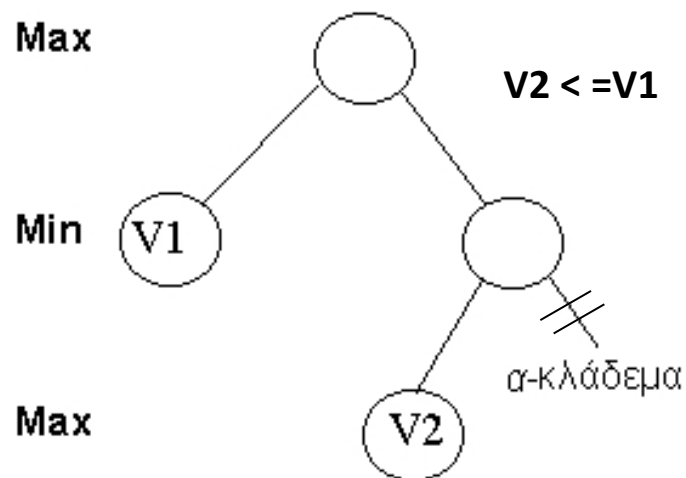


- ✓ Το παιχνίδι αυτό δεν χαρακτηρίζεται δίκαιο αφού έχει αποδειχτεί ότι όποιος παίζει πρώτος σε $4k+1$ ξυλάκια, με $k=1, 2, \dots$ πάντα κερδίζει

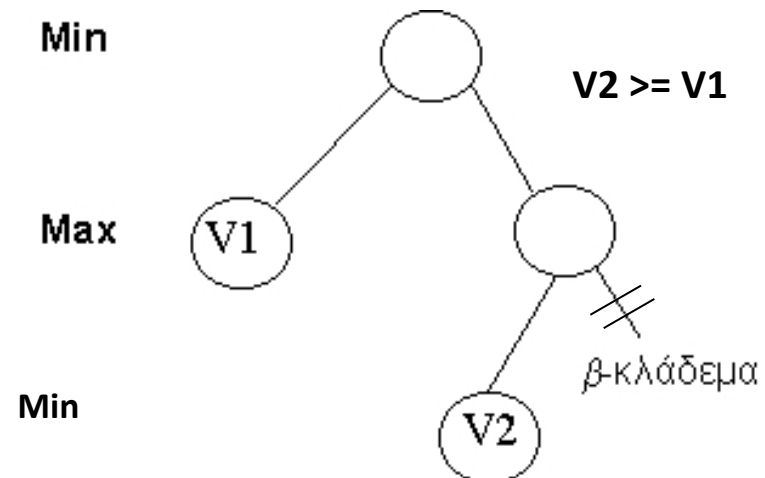


Ο Αλγόριθμος Alpha-Beta

- ❖ Ο αλγόριθμος Άλφα-Βήτα (Alpha-Beta - AB) αποφεύγει την αξιολόγηση καταστάσεων που ικανοποιούν ορισμένες συνθήκες.
- ❖ Ο AB είναι όμοιος με τον Minimax, αλλά με κλάδεμα υποδένδρων, όπως κατά αντιστοιχία ο B&B με τον DFS.
 - ❑ Το κλάδεμα που κάνει ο AB, όπως άλλωστε και ο B&B, **δεν είναι ευρετικό** γιατί βασίζεται σε πραγματικά νούμερα.



$V2 <$ Μεγαλύτερη τιμή του επόμενου επιπέδου



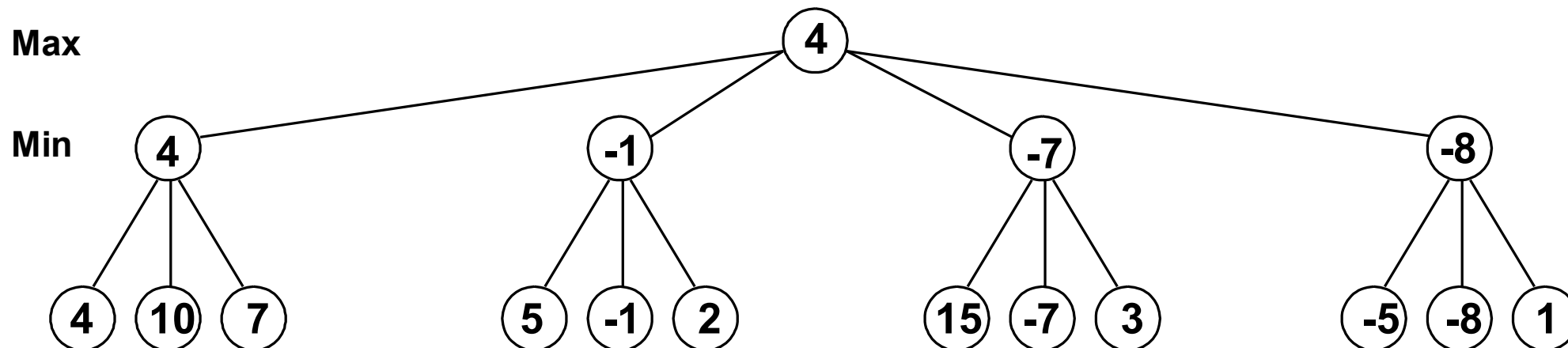
$V2 <$ Μικρότερη τιμή του επόμενου επιπέδου

- ❖ Αρχικά τα όρια είναι $\alpha = -\infty$ και $\beta = +\infty$.
 - ❑ Ο Max προσπαθεί να αυξήσει το α και ο Min να μειώσει το β .



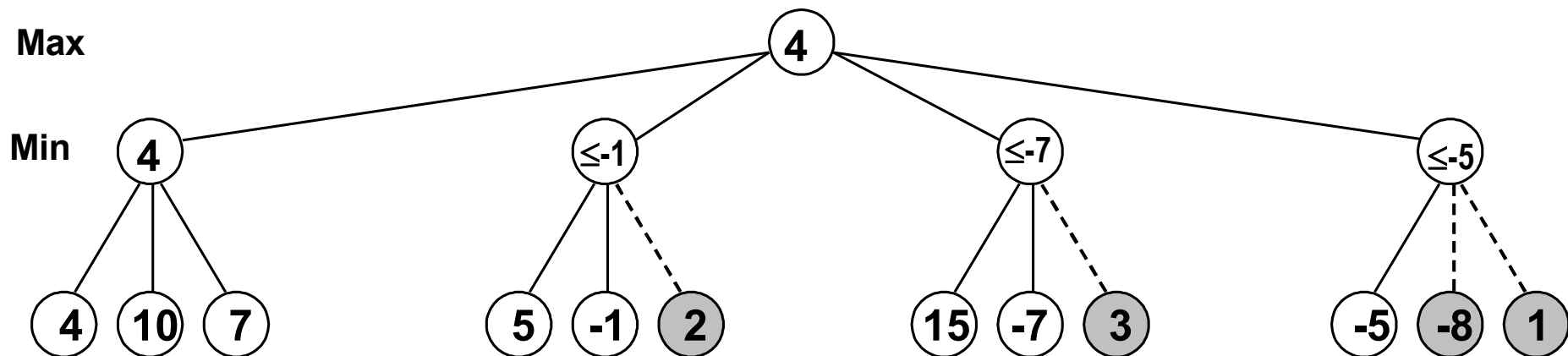
Σύγκριση του AB με τον Minimax (1/2)

Παράδειγμα Minimax





Παράδειγμα Alpha-Beta

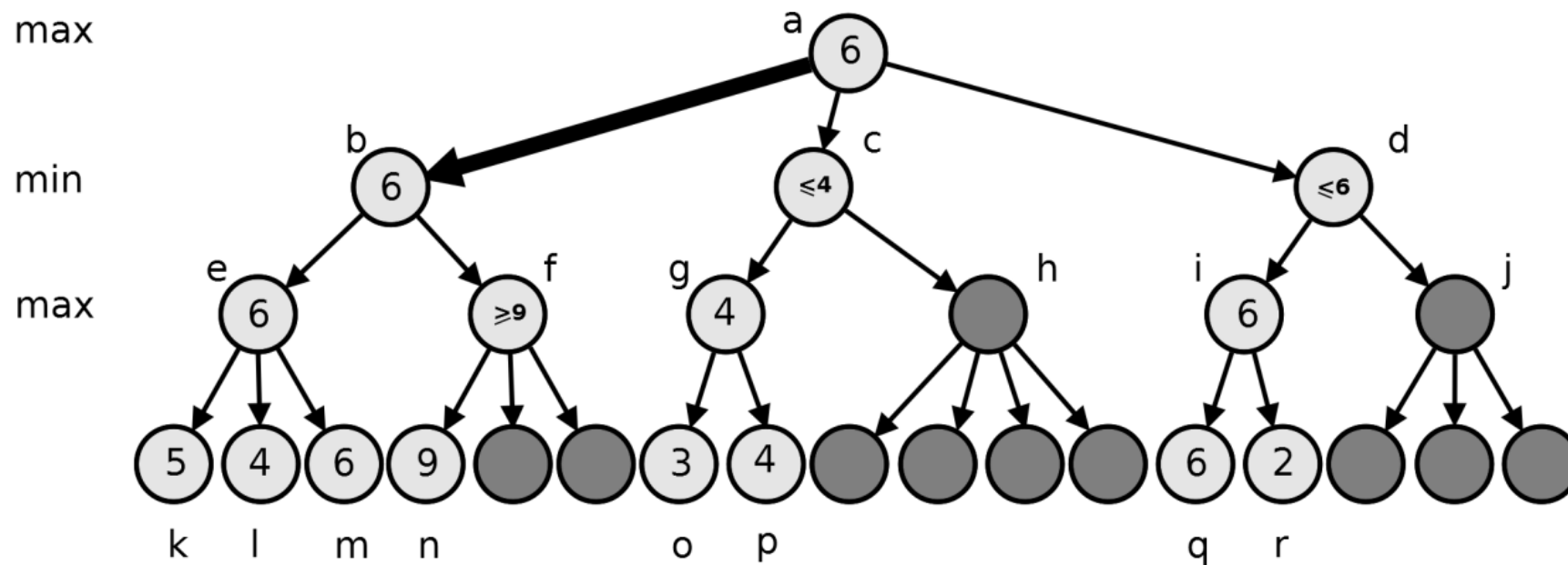


Εξηγήσεις

- ❖ Αξιολογούνται οι πρώτοι αριστεροί κόμβοι (4, 10, 7) και δίνουν την τιμή 4 στον κόμβο πατέρα.
- ❖ Η αξιολόγηση προχωρά στο δεύτερο κλαδί και βρίσκει 5 και -1 οπότε και σταματά **αφού η τιμή του είναι μικρότερη από τη μεγαλύτερη τιμή του ανωτέρου επιπέδου (δηλ. το 4)**, γιατί αν συνεχίζαμε ο επόμενος κόμβος θα είχε τιμή, είτε:
 - ☐ μεγαλύτερη από το -1, που δεν θα επηρέαζε τη διαδικασία καθόσον στον κόμβο πατέρα (min) θα πήγαινε πάλι το -1, ή
 - ☐ μικρότερη από το -1, η οποία και να πήγαινε στον κόμβο πατέρα δεν θα συνέχιζε προς τα επάνω στην ρίζα, αφού η τιμή του θα είναι μικρότερη του 4.
- ❖ Όμοια γίνεται και στους επόμενους κόμβους.



Παράδειγμα





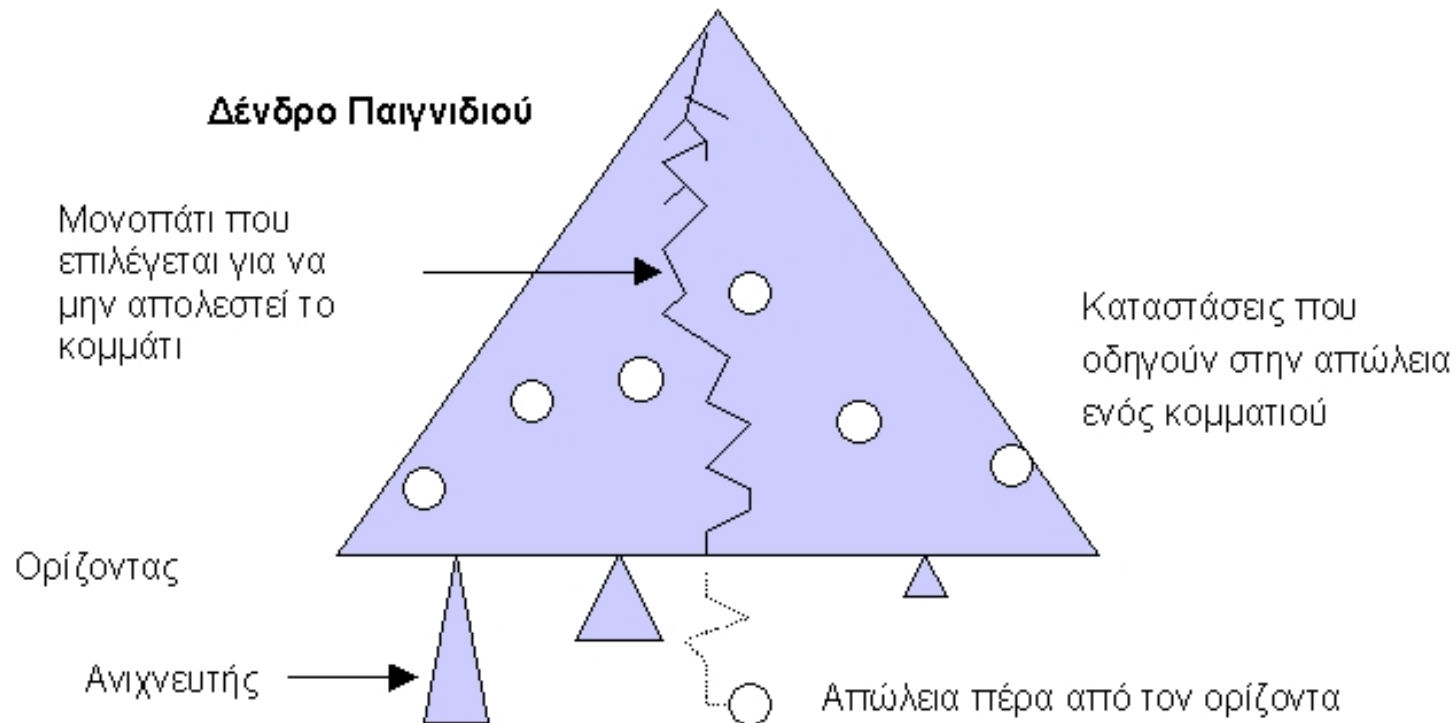
Σύγκριση του AB με τον Minimax (2/2)

- ❖ Κατά προσέγγιση ο AB εξετάζει \sqrt{N} τερματικούς κόμβους, όπου N είναι οι τερματικοί κόμβοι που εξετάζει ο αλγόριθμος Minimax.
- ❖ Η απόδοσή του βελτιώνεται με διάφορες μεθόδους, όπως:
 - ☐ Ευρετικό κλάδεμα του δένδρου παιχνιδιού.
 - ☐ Δυναμική αντί στατικής συνάρτησης αξιολόγησης.
 - ☐ Αποθήκευση τιμών των τερματικών καταστάσεων (transposition tables).
 - ☐ Προκαθορισμένες κινήσεις (χωρίς αναζήτηση) σε αρχικές και τελικές φάσεις του παιχνιδιού (*Openings, End Game moves*).



Το Πρόβλημα του Ορίζοντα

- ❖ Μειονεκτήματα: Η κακή εξέλιξη του παιχνιδιού δεν μπορεί να αποφευχθεί λόγω του συγκεκριμένου βάθους δημιουργίας του δένδρου (ορίζοντα).
 - ❑ Ονομάζεται φαινόμενο του ορίζοντα (horizon effect).



- ❑ Μέρος του προβλήματος επιλύεται όταν η αναζήτηση γίνεται πιο βαθιά σε ορισμένα μόνο σημεία, χρησιμοποιώντας ειδικούς αλγορίθμους που ονομάζονται **ανιχνευτές (scouts)**.



Θεωρία Παιγνίων

- ❖ Είδαμε μία ειδική κατηγορία παιγνίων που χαρακτηρίζονται ως **παίγνια δυο αντιπάλων απόλυτης πληροφoρίας**.
 - ❑ Γενικότερα όμως, η **Θεωρία Παιγνίων** (**Game Theory**) είναι ο επιστημονικός τομέας ο οποίος ασχολείται με τη λήψη αποφάσεων, από τους παίκτες.
 - ❑ **Παίγνιο** είναι μία τυπική αναπαράσταση μιας κατάστασης στην οποία ένας αριθμός από παίκτες αλληλοεπιδρούν σε ένα περιβάλλον που ο καθένας είναι στρατηγικά ανεξάρτητος.
 - ❑ Θεωρούμε ότι οι παίκτες έχουν **ορθολογική συμπεριφορά** (*rational behaviour*) (σε αντίθεση με την συναισθηματική ή μη σώφρονα συμπεριφορά) και ότι ο καθένας θέλει να επιτύχει το καλύτερο αποτέλεσμα που μπορεί.
 - ❑ Οι ενέργειες που αντιστοιχούν σε κάθε παίκτη ονομάζεται και **στρατηγικές** (*strategies*).
 - ❑ Οι παίκτες αυτοί ονομάζονται γενικά και **πράκτορες** (*agents*).
 - ❑ Σε ένα παίγνιο, η κάθε στρατηγική που έχει ένας παίκτης έχει ένα όφελος ή αλλιώς **απολαβή** (*payoff*).
 - ❑ Ένας απλός τρόπος για να αναπαραστήσει τα οφέλη των παικτών ανάλογα με τις ενέργειες που κάνουν είναι ο λεγόμενος **πίνακας απολαβών** (*payoff matrix*).



Παράδειγμα

- ❖ Κλασικό παράδειγμα στη θεωρία παιγνίων είναι το *δίλημμα του φυλακισμένου* (*prisoner's dilemma*).
- ❖ Η Μαρίνα και η Ιωάννα έχουν συλληφθεί για κλοπή και κρατούνται στην αστυνομία σε δύο χωριστά δωμάτια που δεν επικοινωνούν μεταξύ τους.
- ❖ Η αστυνομία προσφέρει σε κάθε μία από αυτές ξεχωριστά την εξής συμφωνία:
 - ☐ Εάν και οι δύο ομολογήσετε την κλοπή τότε θα πάτε 5 χρόνια φυλακή η καθεμία.
 - ☐ Εάν ομολογήσεις εσύ αλλά η άλλη κρατούμενη αρνηθεί τότε θα είσαι ελεύθερη και η συνεργάτιδά σου θα πάει δέκα χρόνια φυλακή.
 - ☐ Εάν και οι δύο αρνηθείτε ότι κάνατε την κλοπή τότε θα πάτε ένα χρόνο φυλακή η καθεμία".

		Μαρίνα	
		Ομολογεί	Αρνείται
Ιωάννα	Ομολογεί	-5, -5	0, -10
	Αρνείται	-10, 0	-1, -1

❖ Ισορροπία Nash



Ισορροπία Nash

- ❖ Ένα προφίλ ενέργειας είναι ισορροπία Nash όταν κανένας πράκτορας δεν μπορεί να πετύχει κάτι καλύτερο επιλέγοντας μια διαφορετική από αυτήν την ενέργεια.
- ❖ «Κλειδώνει» τους πράκτορες σε ένα προφίλ ενέργειας από το οποίο κανείς δεν έχει κίνητρο να αποκλίνει μεμονωμένα.

		Μαρίνα	
		Ομολογεί	Αρνείται
Ιωάννα	Ομολογεί	-5, -5	0, -10
	Αρνείται	-10, 0	-1, -1



Δυοπώλιο

- ❖ Δύο εταιρίες παράγουν το ίδιο αγαθό, για το οποίο μπορούν να χρεώσουν είτε μια Υψηλή είτε μια Χαμηλή τιμή. Κάθε εταιρία επιθυμεί να μεγιστοποιήσει τα κέρδη της.
- ❖ Αν και οι δύο επιλέξουν υψηλή τιμή, το κέρδος και των δύο θα είναι από 1.000 ευρώ.
- ❖ Αν και οι δύο επιλέξουν χαμηλή τιμή, το κέρδος τους θα είναι 600 ευρώ.
- ❖ Αν η μία επιλέξει υψηλή τιμή και η άλλη χαμηλή, τότε αυτή που θα διαλέξει την υψηλή θα έχει ζημία 200 ευρώ, ενώ αυτή με την χαμηλή θα έχει κέρδη 1.200 ευρώ (λόγω περισσότερων πωλήσεων)

		Εταιρία 2	
		Υψηλή	Χαμηλή
Εταιρία 1	Υψηλή	1000, 1000	-200, 1200
	Χαμηλή	1200, -200	600, 600



❖ Ισορροπία Nash

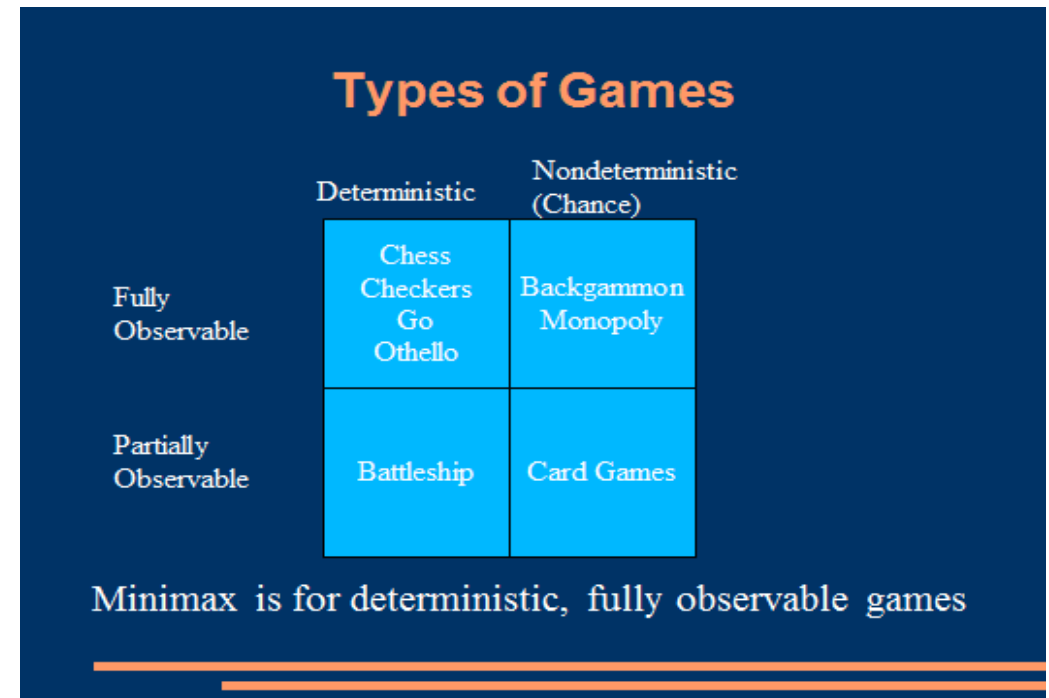
		Εταιρία 2	
		Υψηλή	Χαμηλή
Εταιρία 1	Υψηλή	1000, 1000	-200, 1200
	Χαμηλή	1200, -200	600, 600



Τύποι Παιγνίων

❖ Τα παίγνια χαρακτηρίζονται ως:

- ❑ Παίγνια 2 ατόμων (*two-player*) ή παίγνια πολλαπλών ατόμων (*multiple-player*).
- ❑ Ντετερμινιστικά (*deterministic*) ή στοχαστικά παίγνια (*stochastic*),
- ❑ Συνεργατικά (*cooperative*) ή μη-συνεργατικά παίγνια (*non-cooperative*),
- ❑ Παίγνια στα οποία ο κάθε παίκτης κάνει μία κίνηση κάθε φορά (*turn-taking*) ή παίγνια που γίνονται πολλαπλές κινήσεις κάθε φορά προτού παίξει ο άλλος παίκτης (*simultaneous moves*),
- ❑ Παίγνια στα οποία οι συνολικές απολαβές για τους 2 παίκτες είναι μηδέν (*zero sum*) ή
- ❑ Παίγνια στα οποία οι συνολικές απολαβές και για τους 2 παίκτες δεν είναι μηδέν (*non-zero sum*),
- ❑ Παίγνια στα οποία οι παίκτες έχουν πλήρη πληροφορία για την κατάσταση του προβλήματος (*perfect information*) ή παίγνια στα οποία υπάρχει ατελής πληροφορία (*imperfect information*).





Ερωτήσεις

- ❖ Σε παίγνια δυο αντιπάλων, οι αλγόριθμοι αναζήτησης κατασκευάζουν ένα δένδρο παιχνιδιού (game tree)
- ❖ Σε ένα δένδρο παιχνιδιού με N τερματικούς κόμβους, ο Minimax εξετάζει N κόμβους και ο AB εξετάζει κατά προσέγγιση \sqrt{N} κόμβους.

Κυκλώστε το αντίστοιχο γράμμα Σ (ωστό) – Λ (άθος) στις επόμενες ερωτήσεις: (Προσοχή Σε αυτό το θέμα υπάρχει αρνητική βαθμολογία)

Η αποκοπή που εκτελεί ο αλγόριθμος Alpha-Beta γίνεται με ευριστικό τρόπο

Σ



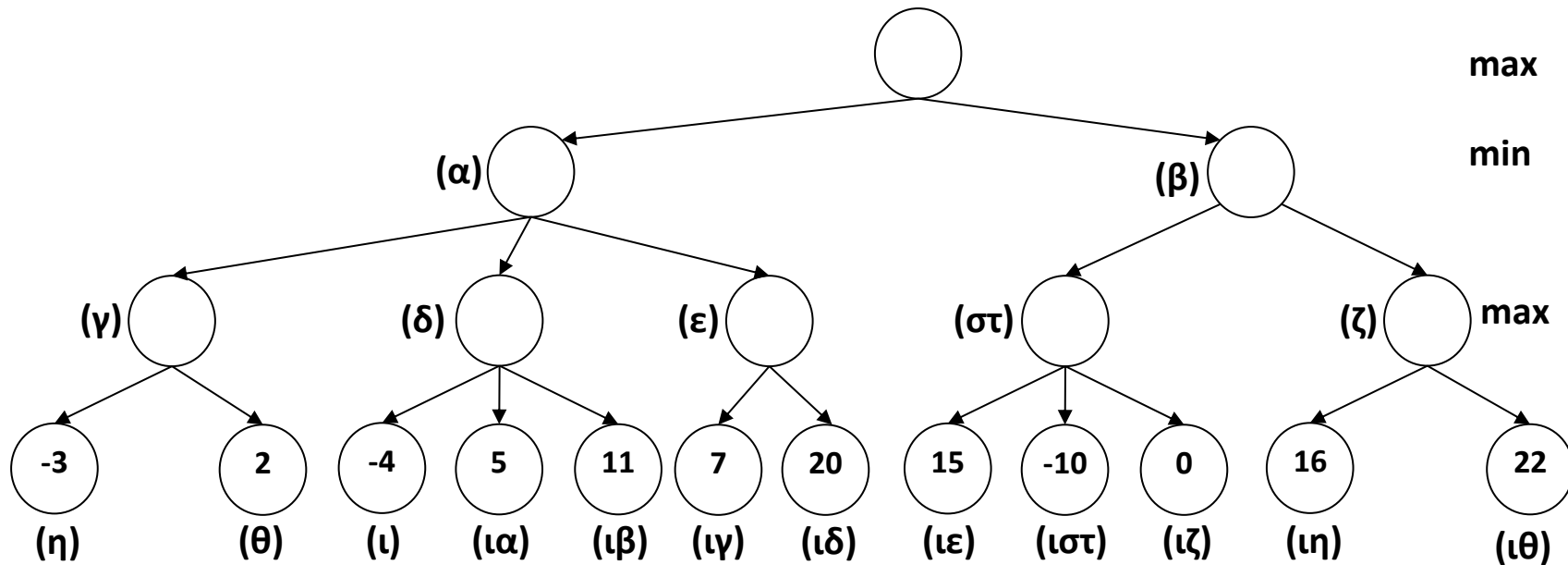
Ο αλγόριθμος MiniMax εγγυάται την πιο συμφέρουσα εξέλιξη ακόμη και όταν ο αντίπαλος κάνει τις ιδανικές κινήσεις

Σ

Λ

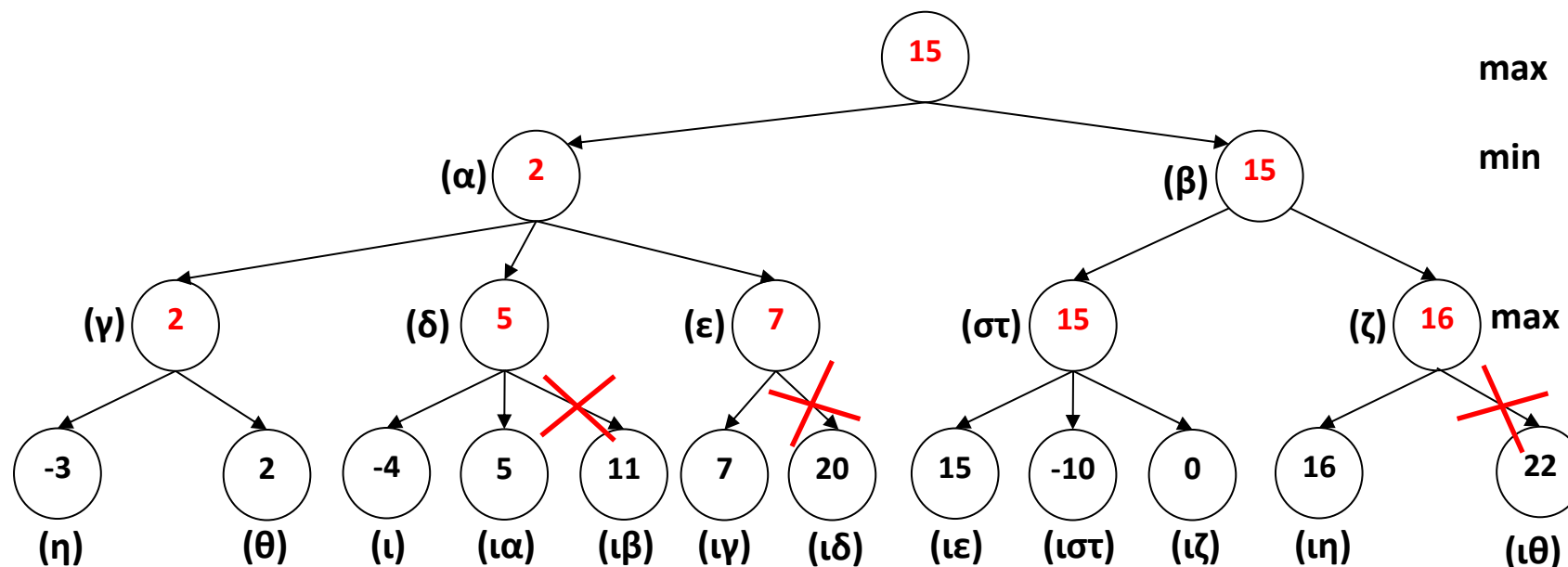
Ασκήσεις

- ❖ 5.1 Το παρακάτω σχήμα παρουσιάζει το δένδρο αναζήτησης ενός παιχνιδιού δύο ατόμων, στο οποίο οι καταστάσεις-φύλλα έχουν αξιολογηθεί από κάποια ευρετική συνάρτηση.
- ☐ Να εφαρμόσετε τον αλγόριθμο Alpha Beta για να συμπληρώσετε τους βαθμούς των υπόλοιπων καταστάσεων.
 - ☐ Να δείξετε με ένα Χ στο σχήμα ποιες καταστάσεις κλαδεύονται από τον αλγόριθμο.
 - ☐ Ποια κατάσταση θα επιλέξει τελικά ο αλγόριθμος (α έως ιθ):





❖ Λύση 5.1:



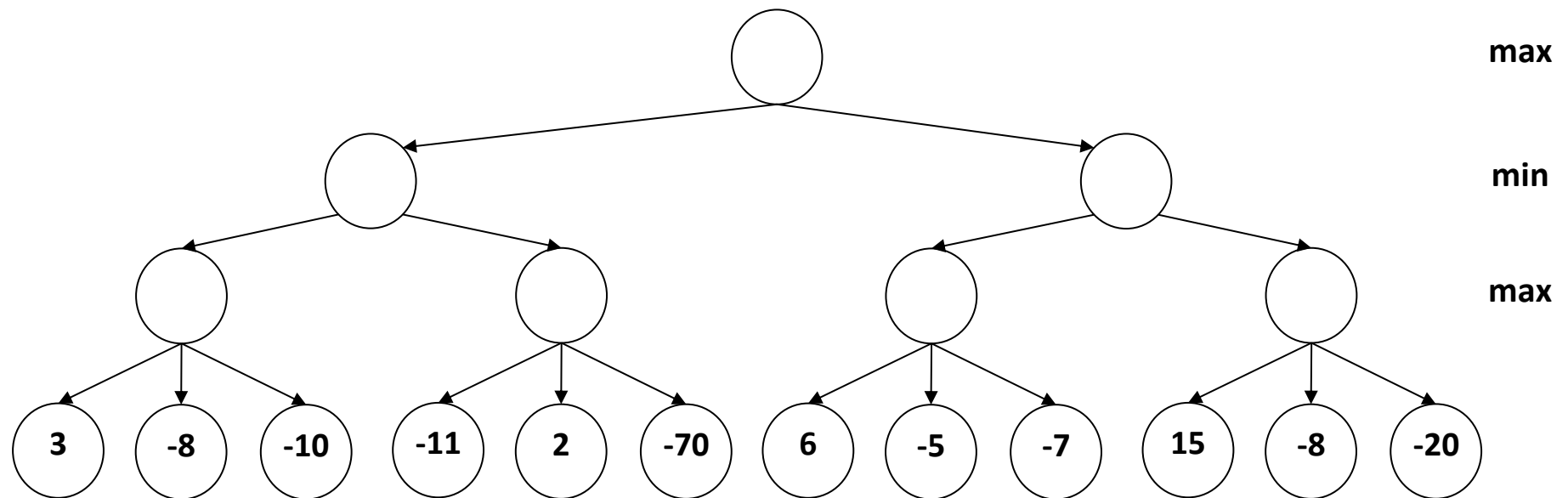
γ) Ποια κατάσταση θα επιλέξει τελικά ο αλγόριθμος (α έως ιθ):**β**.....

- ✓ Μπορείτε επίσης να πειραματιστείτε με το εξής [Πρόγραμμα](#) και το [demo](#)
- ✓ Επιπλέον παραδείγματα [εδώ](#) και ένας [simulator](#) [εδώ](#).



❖ 5.2: Το παρακάτω σχήμα παρουσιάζει το δένδρο αναζήτησης ενός παιχνιδιού δύο ατόμων, στο οποίο οι καταστάσεις-φύλλα έχουν αξιολογηθεί από κάποια ευρετική συνάρτηση.

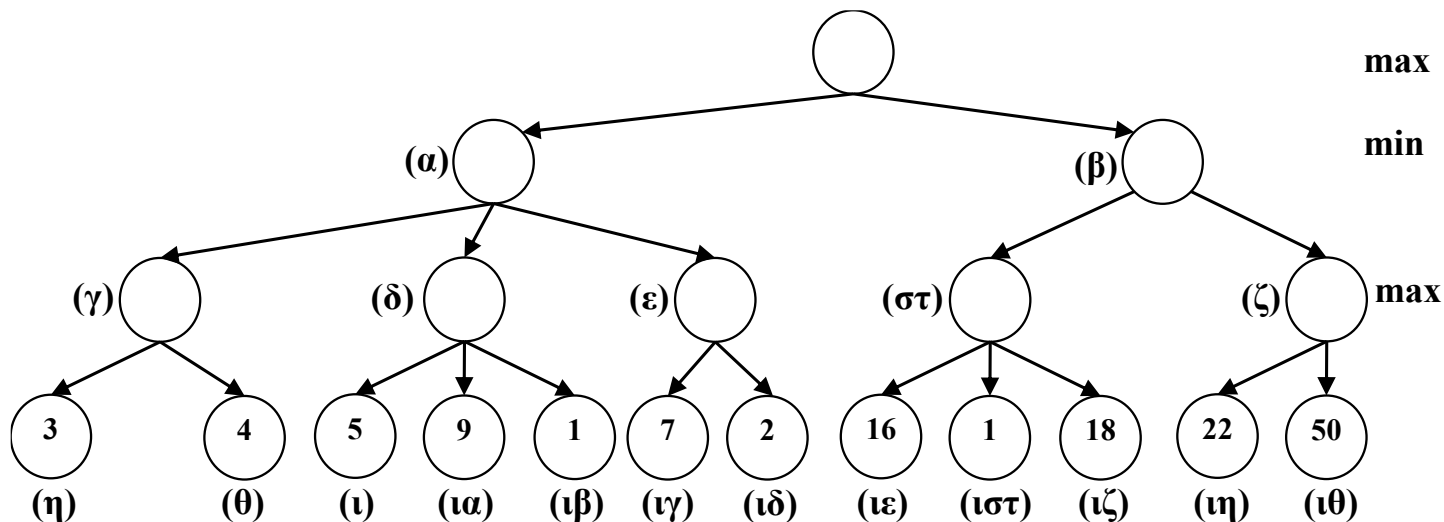
- ☐ Να εφαρμόσετε τον αλγόριθμο Alpha Beta για να συμπληρώσετε τους βαθμούς των υπόλοιπων καταστάσεων.
- ☐ Να δείξετε με ένα X στο σχήμα ποιες καταστάσεις κλαδεύονται από τον αλγόριθμο.
- ☐ Κυκλώστε την κατάσταση που θα επιλέξει τελικά ο αλγόριθμος





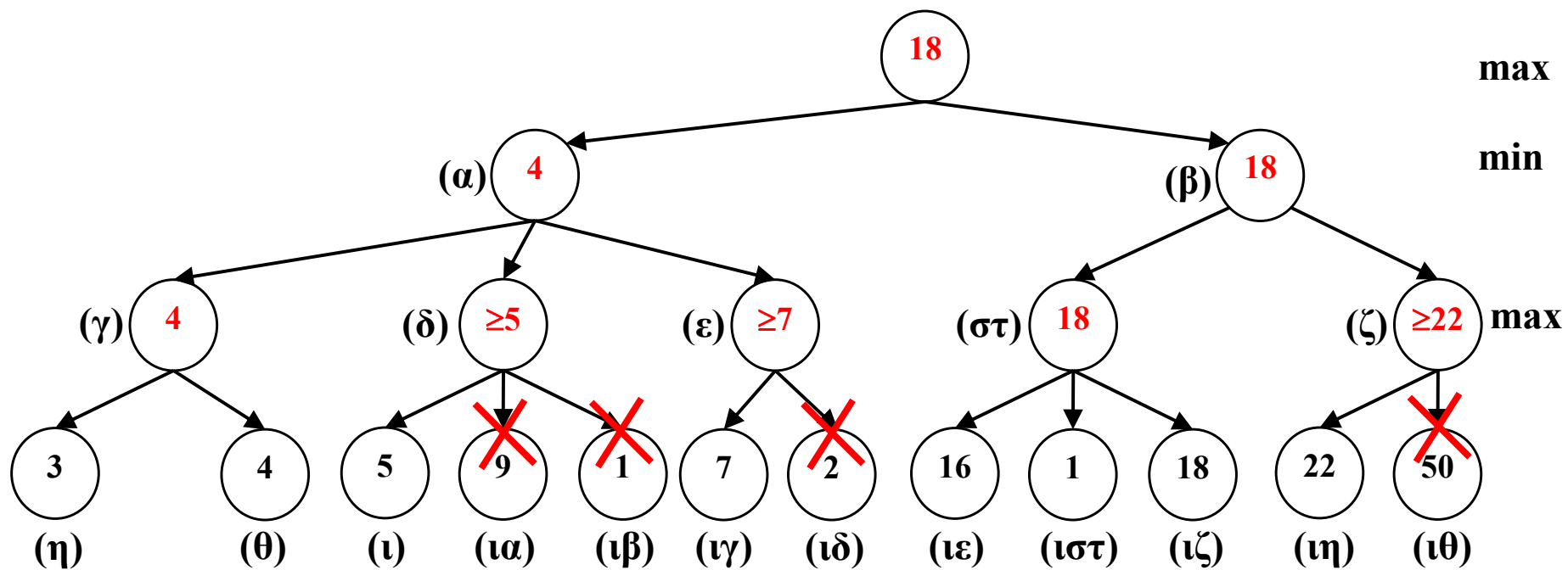
❖ 5.3. Το παρακάτω σχήμα παρουσιάζει το δένδρο αναζήτησης ενός παιχνιδιού δύο ατόμων, στο οποίο οι καταστάσεις-φύλλα έχουν αξιολογηθεί από κάποια ευρετική συνάρτηση.

- ☐ Να εφαρμόσετε τον αλγόριθμο Alpha Beta για να συμπληρώσετε τους βαθμούς των υπόλοιπων καταστάσεων.
- ☐ Δείξτε με ένα X στο σχήμα ποιες καταστάσεις κλαδεύονται από τον αλγόριθμο.
- ☐ Ποια κατάσταση θα επιλέξει τελικά ο αλγόριθμος (α έως ιθ):





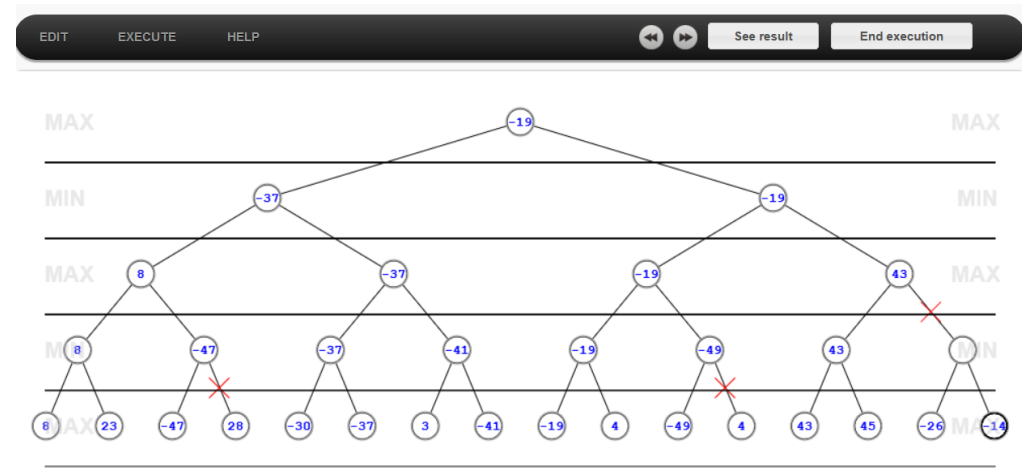
❖ Λύση:



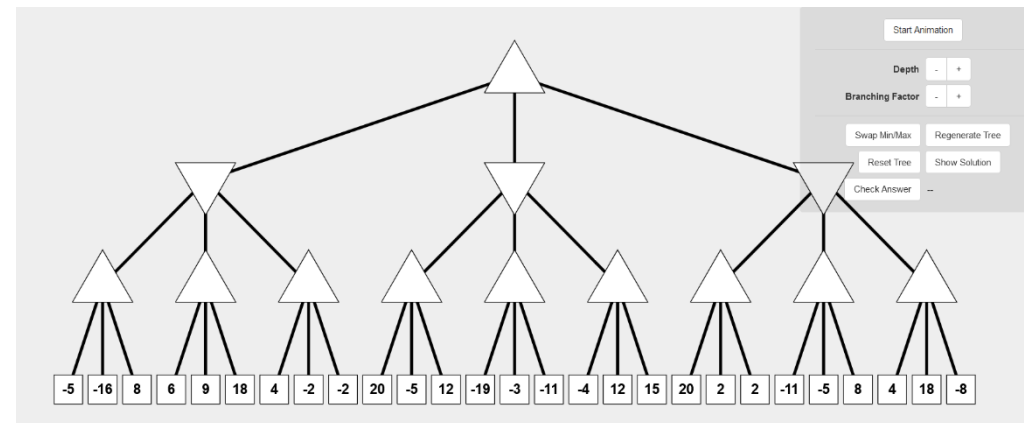


Εξάσκηση

Online AB simulator: https://raphsilva.github.io/utilities/minimax_simulator/



Alpha-Beta Pruning Practice: <https://pascscha.ch/info2/abTreePractice/>





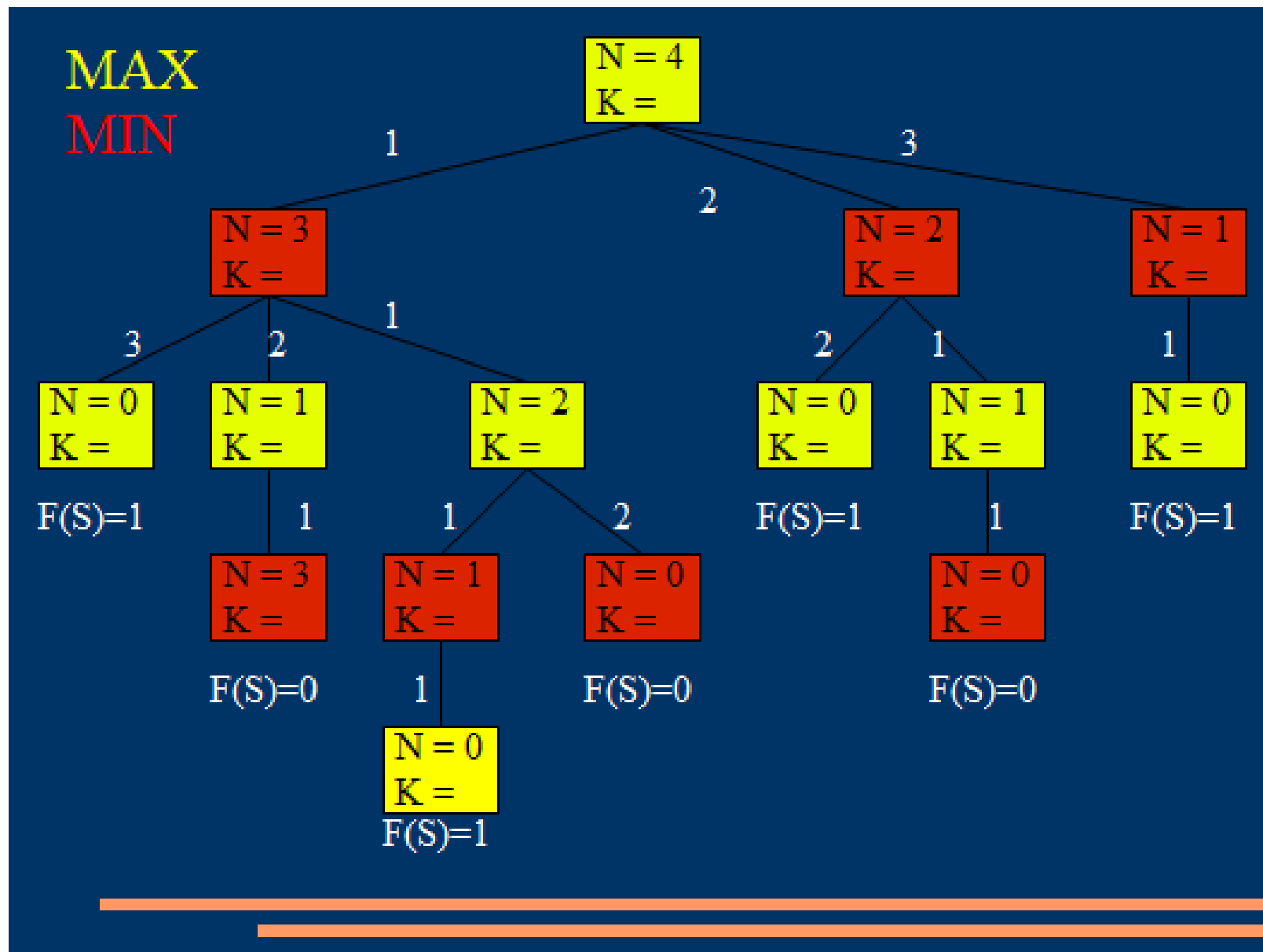
Example

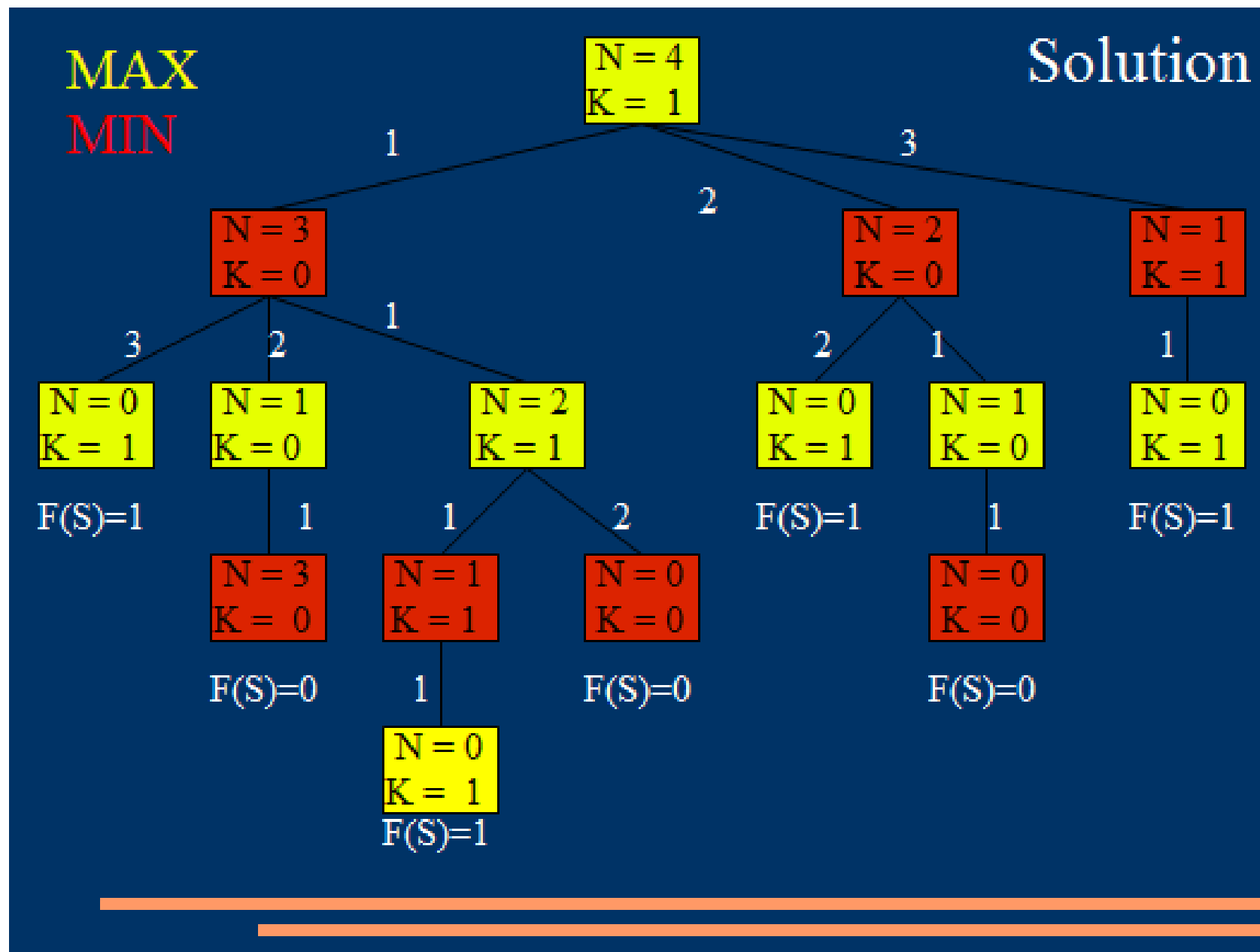
- Coins game
 - There is a stack of N coins
 - In turn, players take 1, 2, or 3 coins from the stack
 - The player who takes the last coin loses



Coins Game: Formal Definition

- Initial State: The number of coins in the stack
- Operators:
 1. Remove one coin
 2. Remove two coins
 3. Remove three coins
- Terminal Test: There are no coins left on the stack
- Utility Function: $F(S)$
 - $F(S) = 1$ if MAX wins, 0 if MIN wins







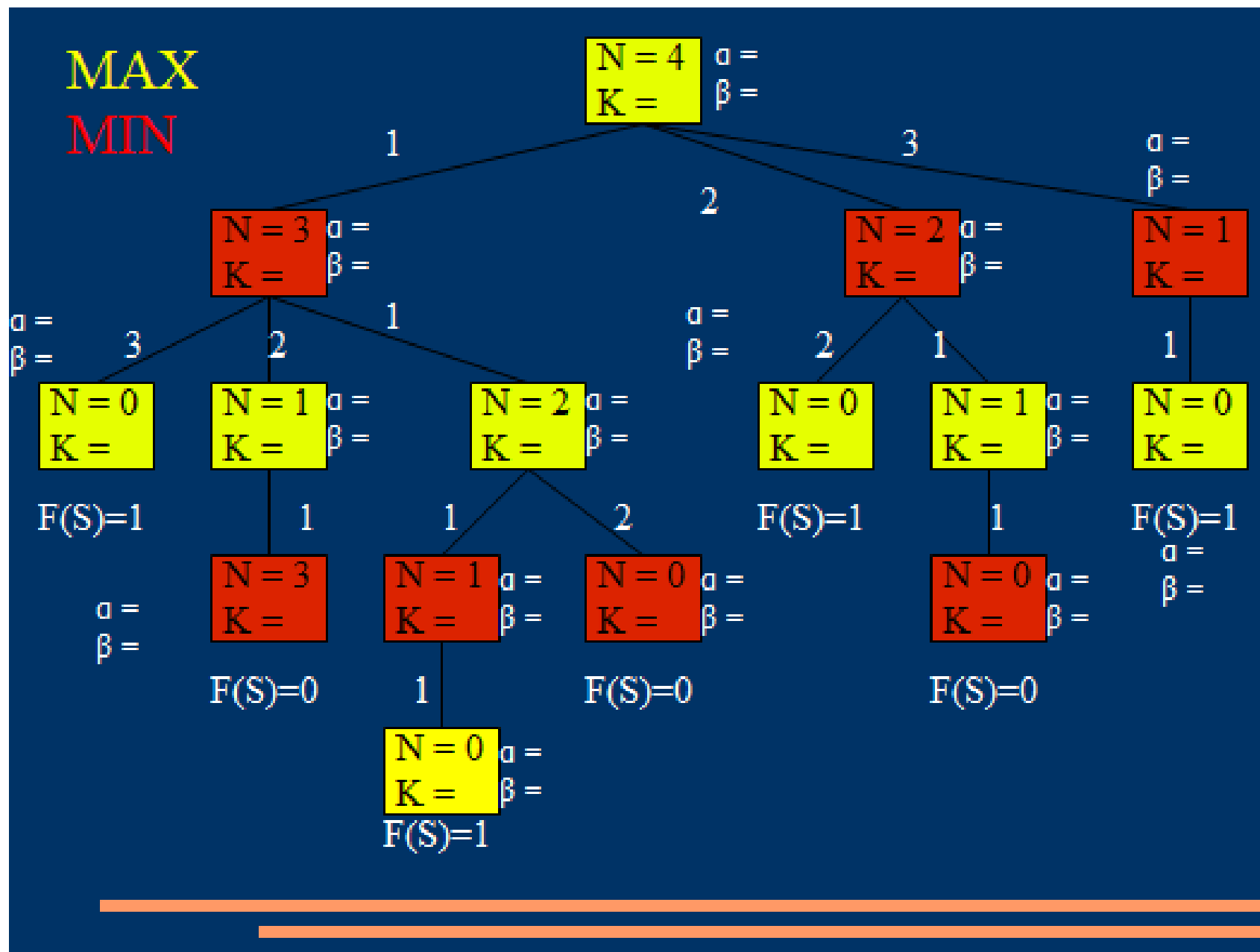
Analysis

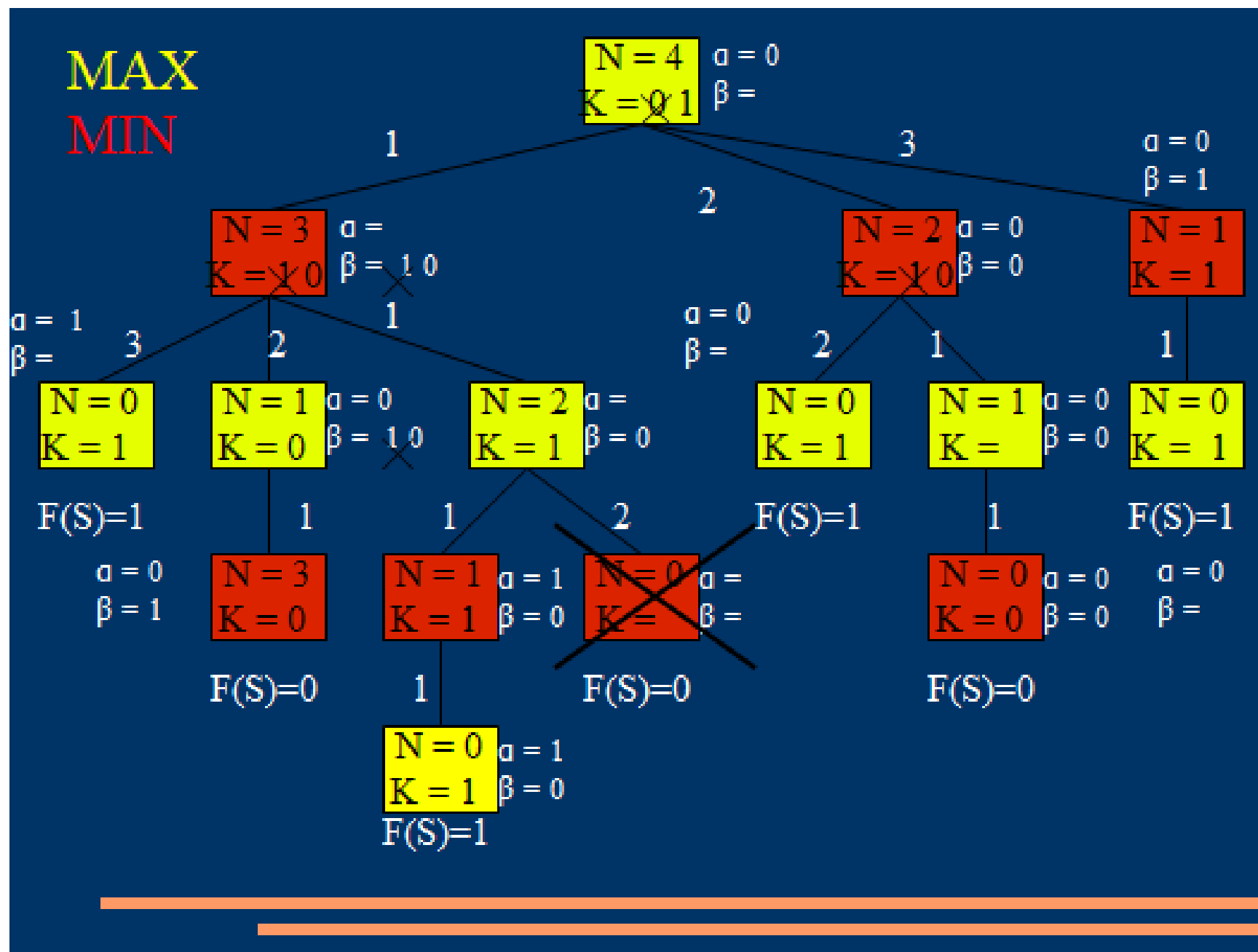
- Max Depth: 5
- Branch factor: 3
- Number of nodes: 15
- Even with this trivial example, you can see that these trees can get very big
 - Generally, there are $O(b^d)$ nodes to search for
 - Branch factor b : maximum number of moves from each node
 - Depth d : maximum depth of the tree
 - Exponential time to run the algorithm!
 - How can we make it faster?



Alpha-Beta Pruning

- Main idea: Avoid processing subtrees that have no effect on the result
- Two new parameters
 - α : The best value for MAX seen so far
 - β : The best value for MIN seen so far
- α is used in MIN nodes, and is assigned in MAX nodes
- β is used in MAX nodes, and is assigned in MIN nodes







Nondeterministic Games

- Minimax can also be used for nondeterministic games (those that have an element of chance)
- There is an additional node added (Random node)
- Random node is between MIN and MAX (and vice versa)
- Make subtrees over all of the possibilities, and average the results