Τεχνητή Νοημοσύνη 1 - Χειμερινό 2020-2021
Εργασια Δεύτερη
Ροβιθάκης Ιωάννης | sdi1800164

Πρόβλημα 1:

i) Εξ ορισμού, **μη βέλτιστος** MIN, είναι αυτός ο οποίος **τουλάχιστον μια φορά δεν λαμβάνει την βέλτιστη απόφαση** - αυτή που προσφέρει, στην περίπτωση του MIN, ελάχιστη δυνατή χρησιμότητα για τον MAX. Αντίθετα, ένας **βέλτιστος** MIN **λαμβάνει πάντα την βέλτιστη απόφαση** και κατα συνέπεια επιτυγχάνει πάντα την ελάχιστη δυνατή χρησιμότητα για τον MAX.

Χωρίς βλάβη γενικότητας για τυχαίο δέντρο παιχνιδιού:

Έστω λοιπον ακολουθίες n κινήσεων του MIN:

Βέλτιστος: **Oi** = m1 m2 m3 ... **mo** ... mn

Μη Βέλτιστος: **Si** = m1 m2 m3 ... **ms** ... mn

Στην περίπτωση αυτή, η Si είναι μη βέλτιστη καθώς διαφέρει από την βέλτιστη μόνο στην κίνηση **ms.**(= μόνο μία μη βέλτιστη απόφαση = base case αφού είναι η ελάχιστη δυνατή διαφορά αποφάσεων)

Έστω λοιπόν ότι η ακολουθία **Oi προσφέρει χρησιμότητα O** στον MIN, η οποία είναι βέλτιστη, αφού παράχθηκε από βέλτιστο MIN.

Αφού γνωρίζουμε ότι η κίνηση ms δεν είναι βέλτιστη συνεπάγεται ότι η χρησιμότητα της για τον MAX θα είναι **μεγαλύτερη** απο την χρησιμότητα που θα πρόσφερε στον MAX η βέλτιστη απόφαση.

Άρα η ακολουθία **Si θα προσφέρει χρησιμότητα S** η οποία λόγω των παραπάνω παρατηρήσεων θα είναι **μεγαλύτερη** της **O**.

Με βάση τα παραπάνω και χωρίς βλάβη της γενικότητας, όποια άλλη περίπτωση "διαφωνίας αποφάσεων" μεταξύ βέλτιστου και μη βέλτιστου MIN λάβουμε υπ όψην μας, θα συνειδητοποιήσουμε ότι όσο περισσότερες μη βέλτιστες αποφάσεις λαμβάνονται, τόσο μεγαλύτερη θα είναι τη αντίστοιχη χρησιμότητα **S** από την βέλτιστη.

Συνεπώς, τελικά κάθε μη βέλτιστη λύση, θα παράγει για τον MAX χρησιμότητα S, πάντα μεγαλύτερη της βέλτιστης για τον MIN χρησιμότητας και κατα συνέπεια ισχύει ο ισχυρισμός της εκφώνησης. => Η χρησιμότητα για τον MAX ενάντια σε μη βέλτιστο MIN, είναι πάντα μεγαλύτερη (=ποτέ μικρότερη) από τη χρησιμότητα που θα είχε ο MAX ενάντια σε βέλτιστο MIN.

(Εναλλακτικά μπορεί να αποδειχτεί με Επαγωγή:

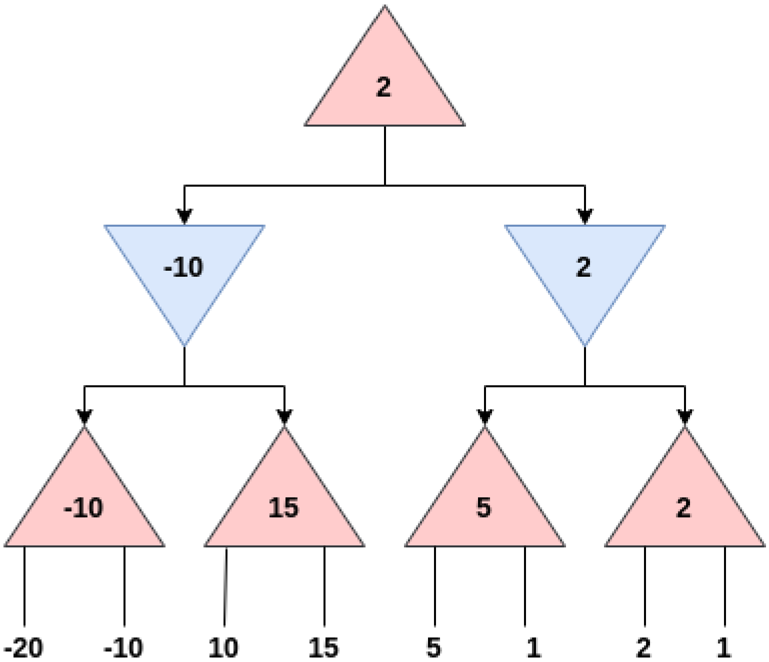
> Επαγ βαση: για μια διαφορετική - μη βελτιστη απόφαση ... O>S

> Έστω ότι ισχύει για n διαφορές ... O>S

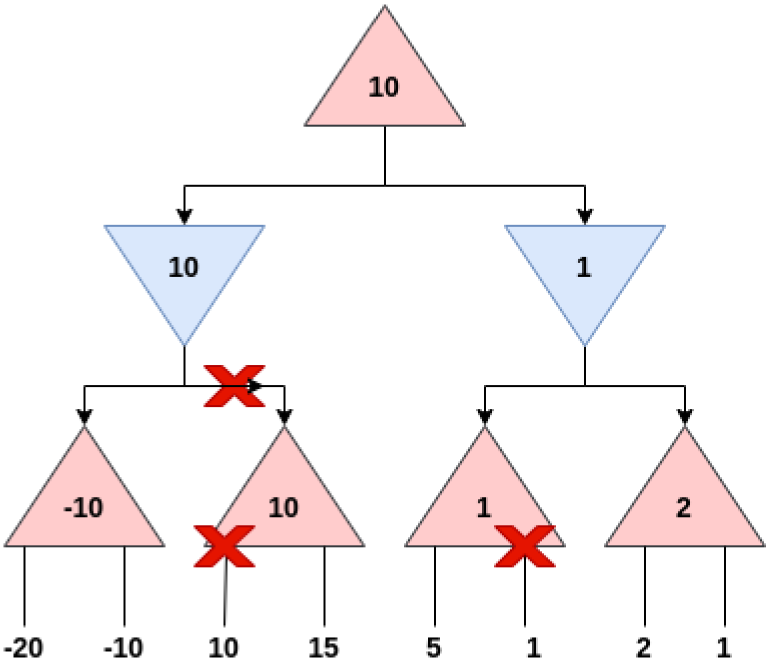
> Θδο ισχυει για n+1 διαφορές ... O>S

(Αφού μόνο ένα μη βέλτιστο στοιχείο αρκέι για να ξεπεράσει το S το O, όσα μη βέλτιστα στοιχεία και να προσθέσουμε θα συνεχίζει να είναι μεγαλύτερο))

ii) Έστω το εξής παράδειγμα με βέλτιστους MIN και MAX:



Για το ίδιο παράδειγμα, αν χρησιμοποιηθούν μη βέλτιστοι MIN, MAX (είτε λόγω constraints σε χωρο/χρόνο, είτε κάποιου interrupt, είτε σχεδιαστικής επιλογής)
θα έχουμε από τον κάθε "παίκτη" τουλαχιστον μια "λάθος" - μη βέλτιστη απόφαση:

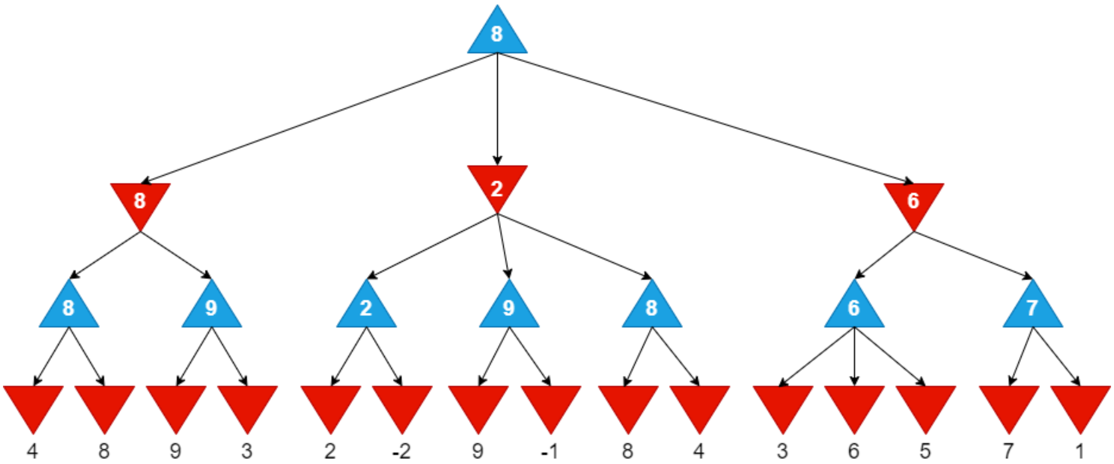


Όπως βλέπουμε, στην περίπτωση αυτή, ενώ ο MAX έχει λάβει 2 μη βέλτιστες αποφάσεις και ο MIN μόνο μια, στην τελική ο μη βελτιστος MAX "νικάει" τον μη βέλτιστο MIN ακόμα καλύτερα και από τον βέλτιστο MAX. Παρόλα αυτά, είναι φανερό οτι αυτό συνέβη στην συγκεκριμένη περίπτωση και ότι κάτι τέτοιο δεν μπορεί να γίνεται με συνέπεια και αξιοπιστία. Με λίγα λόγια, ο MAX στην περίπτωση αυτή απλα στάθηκε τυχερός.

(Τα κόκκινα Χ υποδεικνύουν λάθος-μη βέλτιστες αποφάσεις)

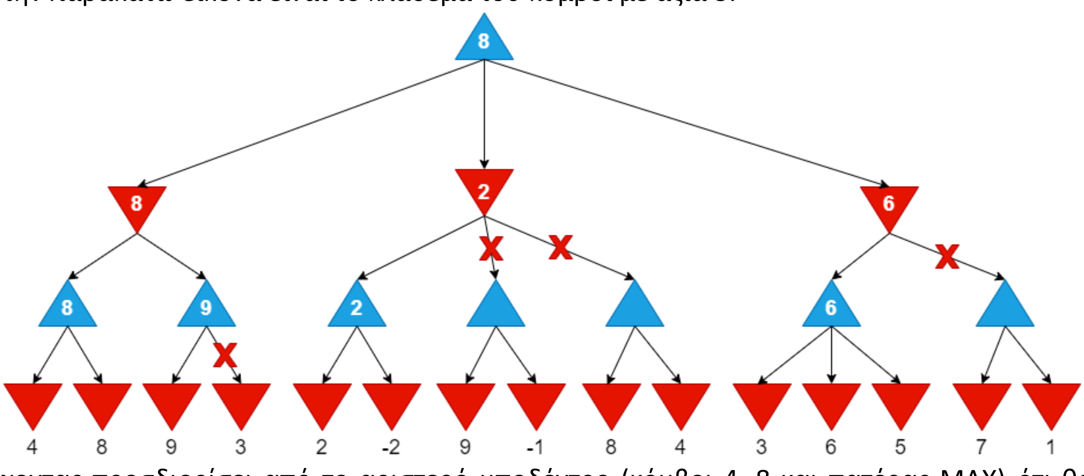
Πρόβλημα 2:

(α) Εκτελώντας τον minimax στο δεδομένο δέντρο θα έχουμε το εξής αποτέλεσμα:



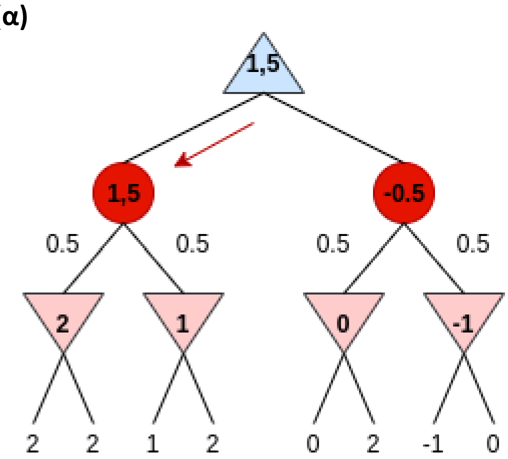
(β) Με βάση την παραπάνω εικόνα, η απόφαση που λαμβάνεται από τον MAX θα είναι να κατέβει στο αριστερό παιδί του, ώστε ο MIN με τη σειρά του να πάει προς το αριστερό του παιδί για να μην επιτρέψει στον MAX να πάρει αξία 9, και τελικά ο MAX να μπορεί να πάει δεξιά , επιτυγχάνοντας την μέγιστη αξία για αυτόν στο δεδομένο δέντρο, δεδομένων των συνθηκών.

(γ) Αν επιθυμήσουμε να βελτιώσουμε τις επιδώσεις που εδωσε ο minimax στο παραπάνω δέντρο, μπορούμε να χρησιμοποιήσουμε τον ALPHA-BETA pruning, ο οποίος "κλαδεύοντας" έξυπνα κάπους κόμβους (και κατά συνέπεια και τους υποκόμβους τους), περιορίζει σημαντικά το συνολικό πλήθος των κόμβων που πρέπει να επισκεφτούμε. Βασιζόμενος στη λογική ότι ο ένας παίκτης πάντα θα διαλέγει την μέγιστη επιλογή, ενώ ο άλλος θα διαλέγει πάντα την ελάχιστη, χρησιμοποιεί μια μορφή άνω/κάτω φραγμάτων για να προσδιορίσει αν ένας κόμβος ή ένα υποδέντρο πρέπει να κλαδευτεί. Ένα παράδειγμα της παραπάνω λογικής στην παρακάτω εικόνα είναι το κλάδεμα του κόμβοι με αξία 3.



Έχοντας προσδιορίσει από το αριστερό υποδέντρο (κόμβοι 4, 8 και πατέρας MAX) ότι θα μπορεί να επιλέξει μια κίνηση που θα οδηγήσει σε αξία 8, ο MIN ελέγχει το δεξί του υποδέντρο για εναλλακτικές, καλύτερες κινήσεις. Ομως, συναντά τον κόμβο με αξία 9, και αφού στο σημείο αυτό επιλέγει κίνηση ο MAX, η κίνηση για αξία 9 θα επιλεγεί σίγουρα εαν δεν βρεθεί κάποια μεγαλύτερη. Συνεπώς είναι προφανές ότι δημιουργείται ένα κάτω φράγμα της αξίας της κίνησης με ελάχιστη αξία 9. Συνεπώς, ο κόμβος 3 κλαδεύεται καθώς ο MIN έχει ήδη διαθέσιμη καλύτερη κίνηση (χαμηλώτερης αξίας) και δεν χρειάζεται να ελέγξει κανέναν άλλο κόμβο αφού υπάρχει το προαναφερθέν κάτω φράγμα. Με αντίστοιχη λογική λειτουργούν και οι υπόλοιπες περιπτώσεις κλαδέματος

Πρόβλημα 3:
(α)



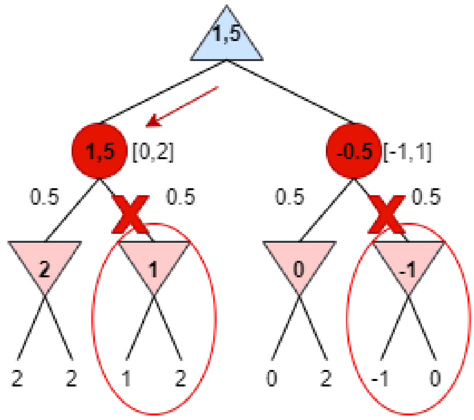
Στην πράξη, η λογική παραμένει ίδια με αυτή του minimax, με μόνη διαφορά ότι προστίθονται "κόμβοι τύχης", στους οποίους η απόφαση που θα παρθεί είναι άγνωστη και εξαρτάται από κάποια πιθανότητα. Στην περίπτωση μας, τα ενδεχόμενα είναι ισοπίθανα, οπότε για να υπολογίσουμε τις τιμές των κόμβων τύχης, αρκεί να υπολογίσουμε τον μέσο όρο των τιμών των διαθέσιμων επιλογών (Εναλλακτικά απλά πολ/ζουμε κάθε τιμή υποκόμβου με την αντίστοιχη της πιθανότητα και προσθέτουμε τα αποτελέσματα). Με βάση τα παραπάνω, καλύτερη κίνηση για τον MAX στο δεδομένο δέντρο, είναι η επιλογή τους αριστερού κόμβου τύχης, καθώς τελικά όδηγεί "πιο συχνα-αξιόπιστα" σε κίνηση που ευνοεί τον MAX.

(β) Αν μας δωθούν μόνο οι 6 πρώτοι κόμβοι, δεν μπορούμε να "κλαδέψουμε" το δέντρο και να μην υπολογίσουμε τους άλλους δύο, διότι λόγω της ύπαρξης του κόμβου τύχης, δεν ξέρουμε με απόλυτη σιγουριά τι τιμές μπορεί να προσφέρει το τελευταίο υποδέντρο. Ειδικά αφού οι τιμές των φύλλων ανοίκουν στο [-inf,+inf], μπορούμε να σκεφτούμε το παράδειγμα, οι αξίες και των δύο κόμβων της τελευταίας επιλογής να είναι +inf, αναγκάζοντας τον MIN να επιλέξει αυτή την τιμή. Η παραπάνω περίπτωση είναι προφανώς μια περίπτωση που αν και απίθανη, υπάρχει, και σε καμια περίπτωση δεν συμφαίρει τον MAX να την αγνοήσει "κλαδεύοντας" την.

Αντίθετα, αν μας δωθούν οι πρώτοι 7 κόμβοι, αφού την πρώτη (= Σε χαμηλότερο επίπεδο) επιλογή, την κάνει ο MIN, τοτέ ανάλογα την περίπτωση μπορούμε να κλαδεύσουμε τον 8ο κόμβο, καθώς η μία αυτή τιμη (η 7η) αρκεί για να θέσει ένα "άνω φράγμα" για την τιμή του υποδέντρου με τη λογική που εξήγησα και παραπάνω για τον alpha-beta. Εδώ όμως ο MIN κανει την επιλογή, οπότε γνωρίζοντας ότι πάντα θα επιλέγει την χαμηλώτερη τιμή, ότι τιμή και να έχει ο 8ος κόμβος, δεν έχει σημασία. Αυτό δεν ισχύει βέβαια εαν η τιμή που υπάρχει στον 7ο κόμβο είναι μεγαλύτερη της "μέχρι εκεί βέλτιστης", οπότε και θα ελεγθεί ο 8ος κόμβος για ύπαρξη ακόμα καλύτερης διαθέσιμης τιμής.

(γ) Αν έχουν αποτιμηθεί μόνο οι τιμές των πρώτων δύο φύλλων, η τιμή του πρώτου κόμβου MIN θα είναι 2 και κατα συνέπεια, προσφέρεται στην αξία του πρώτου κόμβου τύχης 0.5*2=1. Η άλλη πλευρά δεν έχει αποτιμηθεί ακόμα, αλλά γνωρίζουμε ότι οι τιμές των φύλλων της ανοίκουν στο [-2,2]. Συνεπώς ο αντίστοιχος κόμβος MIN θα προσφέρει στον κόμβο τύχης μία τιμή από το -2 μέχρι το 2. Αν προσφέρει -2 τότε θα έχουμε 0.5*-2 = -1, ενώ εάν προσφέρει 2 θα έχουμε 0.5*2=1.
Άρα με βάση την παραπάνω λογική, ο 1ος κόμβος τύχης μπορεί να έχει στην δεδομένη "στιγμή", τιμές στο [1-1,1+1], δηλαδή στο [0,2]

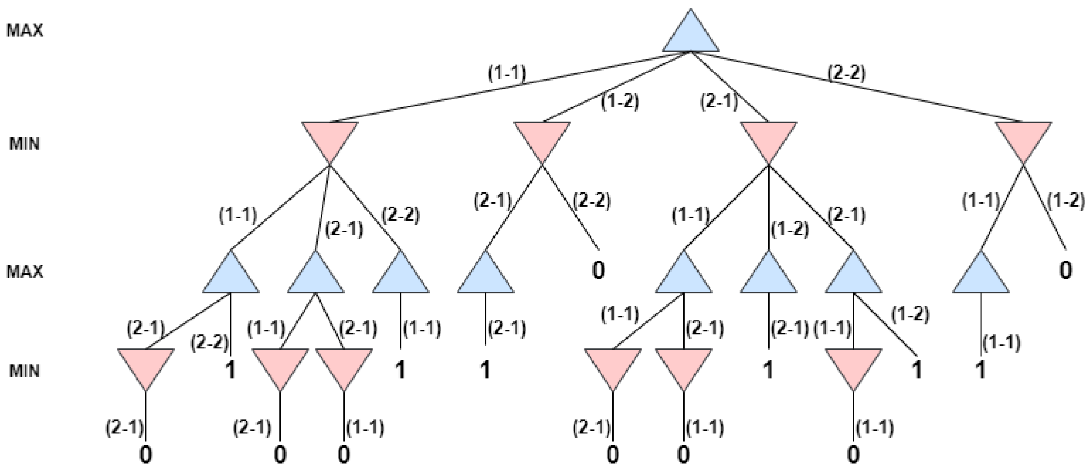
(δ) Λόγω της ύπαρξης κόμβων τύχης, δεν είναι δυνατό να εφαρμόσουμε τον α-β όπως σε προηγούμενες περιπτώσεις, αφού οι κόμβοι που συνδέονται με έναν κοινό κόμβο τύχης, είναι στην πράξη "υποκατάστατοι" ο ένας του άλλου, συνεπώς η επιλογή με βάση πιθανότητα και όχι ντετερμινιστικά, δεν μας επιτρέπει κανονκά να έχουμε ανω/κάτω φράγματα σε κόμβους τύχης. Παρόλα αυτά, αφού υπάρχει στην περίπτωση μας ο περιορισμός τιμών στο διάστημα [-2,2], μπορούμε με την λογική που είδαμε στο παραπάνω ερώτημα να θέσουμε τα δύο εξής φράγματα, και να κλαδέψουμε τους κόμβους που βλέπετε. Ο MAX θα διαλέξει και πάλι την ίδια κίνηση καθώς το διάστημα [0,2] τον συμφαίρει πολύ περισσότερο από το [-1,1]. (**Σημαντική σημείωση:** Το κλάδεμα είναι δυνατό επείδη στην περίπτωση μας υπάρχει το περιορισμένο διάστημα και ταιριάζουν οι αριθμοί του προβλήματος. Γενικά σε προβλήματα με κόμβους τύχης συνήθως δεν μπορούμε να κλαδέψουμε τίποτα και πρέπει να εξερευνήσουμε όλο το δέντρο, πχ αν είχαμε [-inf,+inf])



Πρόβλημα 4:

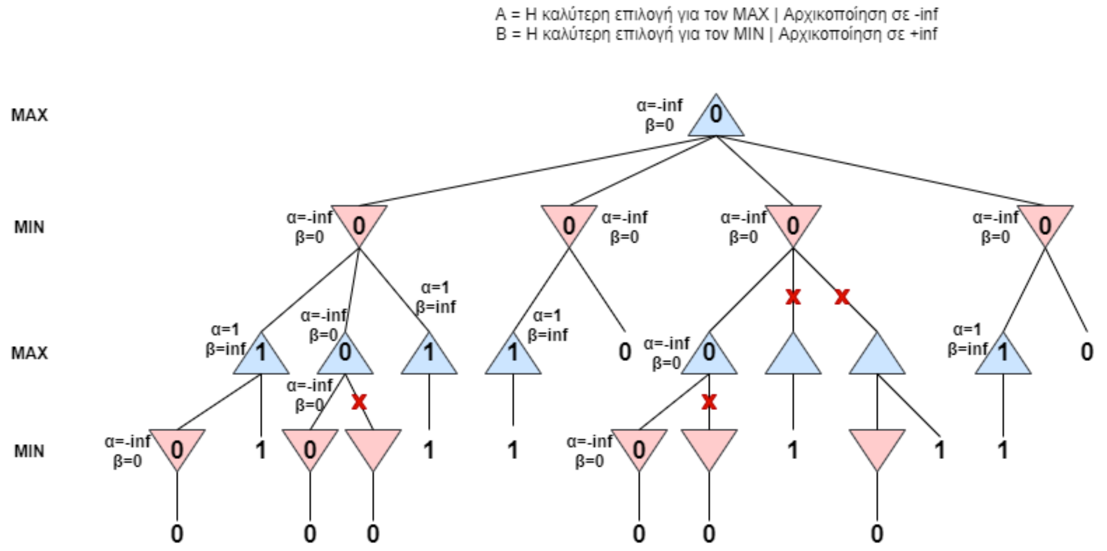
1. NIM : Σε κάθε γύρο, κάθε παίκτης πρέπει με την σειρά του να αφαιρέσει από μία από τις διαθέσιμες στοίβες τουλάχιστον ένα αντικείμενο, μέχρι όσα θέλει (από την ίδια στοίβα). Ο παίκτης που θα αφαιρέσει το τελευταίο αντικείμενο της τελευταίας στοίβας κερδίζει. Συνεπώς, στην περίπτωση που μας δίνεται με 2 στοίβες, 2 αντικειμένων η κάθε μία, το δέντρο του παιχνιδιού θα είναι το εξής:

Επεξήγηση συμβολισμού κινήσεων: (x - y) x = αριθμός στοίβας, y = αριθμός στοιχείων που αφαιρούνται
πχ (1-2) = αφαίρεση 2 στοιχείων από την πρώτη στοίβα



(Παρά την εμφανή συμμετρία επέλεξα να απεικονίσω ολόκληρο το δέντρο του παιχνιδιου καθώς φαίνεται καλύτερα το κλάδεμα, παρόλο που αν χειροστούμε σωστά το πρόβλημα, μπορούμε να το λύσουμε αποκλειστικά με το μισό δέντρο, και να αναπαραστήσουμε το υπόλοιπο έμμεσα, χάρη στη συμμετρία)

2.



Επειδή οι τιμές του δέντρου είναι 0 ή 1 (Δεν υπάρχει μεγάλη διακύμανση => αρκετές ισότητες) επέλεξα τον αβ μου να "κλαδεύει" το δέντρο και σε περίπτωση ισότητας.

Δεν συμπεριέλαβα τις ακριβείς κινήσεις που αντιστοιχούν σε κάθε κόμβο όπως πάνω για να έχω χώρο για τις τιμές των α και β. Ειναι αντίστοιχες με του αριστερού /ή αναλόγως του δεξιού "υποδέντρου" του σχεδιαγράμματος του προηγούμενου ερωτήματος.

Θα μπορούσαμε να απλοοιήσουμε το δέντρο αφαιρώντας τους κόμβους βάθους για τους οποίους υπάρχει διαθέσιμη **μοναδική** κίνηση η οποία οδηγεί συγκεκριμένη τελική κατάσταση, και να τους αντικαταστήσουμε με την ανάλογη τιμή της, για να κάνουμε την αναπαράσταση πιο απλή

Παρατηρούμε ότι κλαδεύεται το μεγαλύτερο τμήμα του συμμετρικού υποδέντρου. Αν είχαμε επιλέξει να αναπαραστήσουμε μόνο το μισό λόγω συμμετρίας, θα κλαδευόταν μόνο ένα κλαδί από το δέντρο (της περίπτωσης που κάθε κίνηση των MIN MAX είναι αφαίρεση ενός μόνο αντικειμένου)

3. Αν και οι δύο παίκτες παίζουν αλάνθαστα (είναι βέλτιστοι) τότε όπως παρατηρούμε από το διάγραμμα του ερωτήματος 1, κερδίζει πάντα ο παίκτης ο οποίος παίζει δεύτερος. Αυτό φαίνεται ξεκάθαρα στο σχήμα του προηγούμενου ερωτήματος. Ο MAX δεν έχει καμία δυνατή επιλογή που να τον οδηγεί σε νίκη. Αν ο ΜΑΞ πάρει 2 αντικείμενα από την μία στήλη, τότε ο MIN νικάει αμέσως παίρνοντας τα δύο που απομένουν στην άλλη στήλη, ενώ αν πάρει μόνο ένα αντικείμενο από μια στήλη, ο MIN αρκεί να πάρει επίσης 1 από την άλλη στηλη, και να αναγκάσει έτσι τον MAX να πάρει πάλι 1 από οποιαδήποτε στήλη, οδηγώντας πάλι τον MIN σε εγγυημένη νίκη. Το ότι οι δύο πράκτορες MAX και MIN λειτουργούν βέλτιστα, μας εγγυάται ότι οι αποφάσεις τους θα είναι οι βέλτιστες δυνατές, δηλαδή ο MIN θα κάνει όλες τις κινήσεις που έχει διαθέσιμες για να μην αφήσει στον MAX κανένα περιθώριο για νίκη. Μάλιστα, με λίγη έρευνα στο διαδίκτυο, μαθαίνουμε ότι το NIM σαν παιχνίδι, έχει ένα σύνολο από καταστάσεις και στρατηγικές οι οποίες μπορούν αν χρησιμοποιηθούν/εφαρμοστούν σωστά να εξασφαλίσουν την νίκη του ενός παίκτη. Η παρούσα περίπτωση αποτελεί ένα πολύ απλό παράδειγμα μίας τέτοια περίπτωσης.

Τα σχήματα γίναν με χρήση του εργαλείου "https://app.diagrams.net/"