Σπανάκης Δημήτριος – sdi1800183

Τεχνητή Νοημοσύνη - Εργασία 2

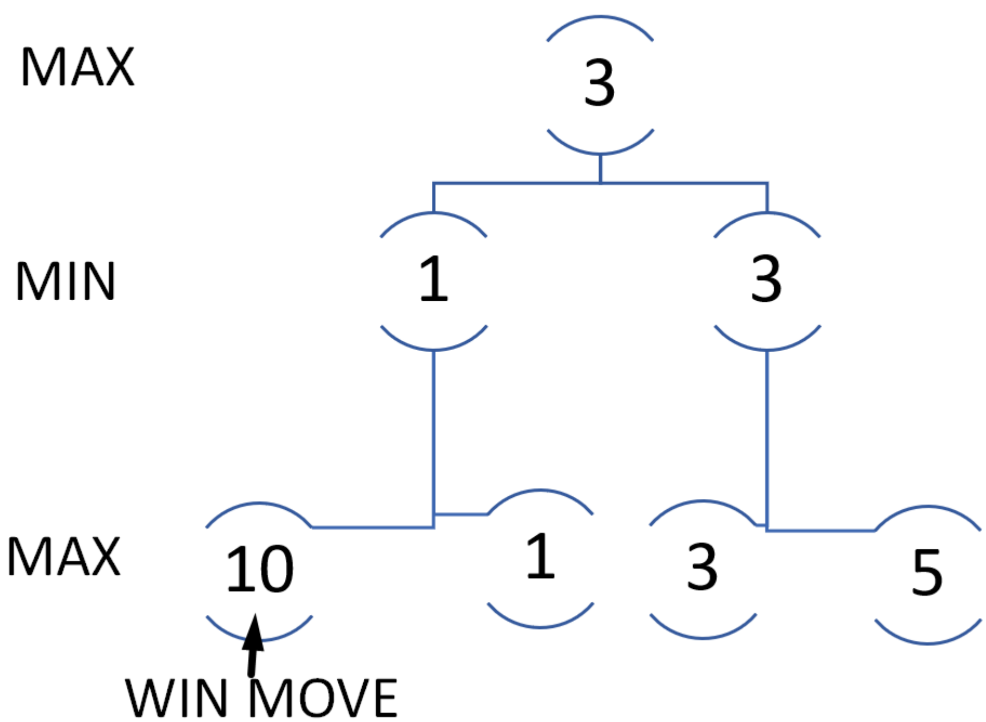
Πρόβλημα 1

Αποδείξτε τον εξής ισχυρισμό. Για κάθε δένδρο παιχνιδιού, η χρησιμότητα για τον MAX που υπολογίζεται χρησιμοποιώντας αποφάσεις minimax εναντίον ενός μη βέλτιστου (suboptimal) MIN δεν είναι ποτέ μικρότερη από την χρησιμότητα που υπολογίζεται παίζοντας εναντίον ενός βέλτιστου MIN.

Ας υποθέσουμε ότι έχουμε ένα MIN κόμβο του οποίου τα παιδιά είναι φύλλα του δέντρου. Αν ο MIN είναι suboptimal, τότε το value του MIN θα είναι μεγαλύτερο η ίσο με το value αν ο MIN ήταν optimal. Έτσι το value του MAX κόμβου ο οποίος θα είναι ο πατέρας του MIN μπορεί να εξισωθεί μόνο προς τα πάνω. Έτσι, η τιμή του κόμβου MIN, αυξάνεται προς την MAX τιμή του πατέρα, μέσω επαγωγικής υπόθεσης. Αυτή η λογική που περιεγράφηκε παραπάνω μπορεί να εφαρμοστεί κατά μήκος όλου του δέντρου.

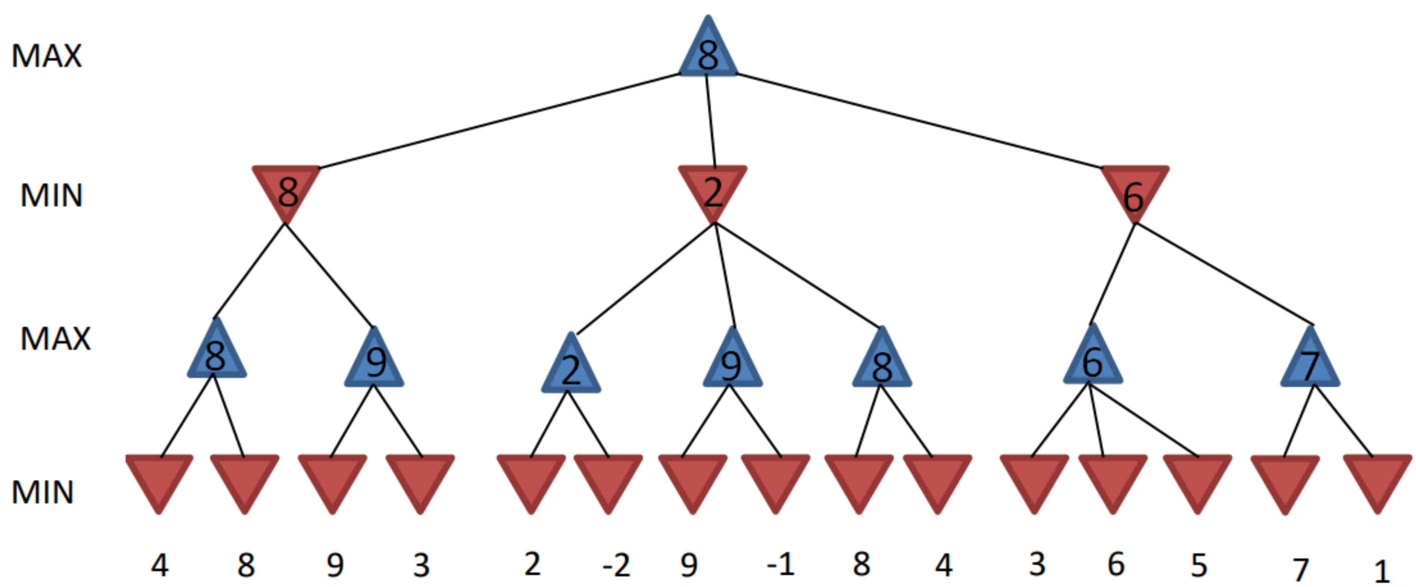
Μπορείτε να βρείτε ένα δένδρο παιχνιδιού στο οποίο ο MAX μπορεί να τα καταφέρει ακόμα καλύτερα χρησιμοποιώντας μια μη βέλτιστη (suboptimal) στρατηγική εναντίον ενός μη βέλτιστου MIN;

Αν μπορούσαμε να προβλέψουμε διάφορες παγίδες σε ένα παιχνίδι που μπορεί να πέσει ο MIN μας, τότε θα μπορούσαμε να σκεφτούμε πως μπορούμε να οδηγήσουμε τον MIN εκεί ώστε να εξασφαλίσουμε την νίκη πιο γρήγορα και πιο σίγουρα. Για παράδειγμα, αν υποθέσουμε ότι το κάτω δέντρο παιχνιδιού έχει για values τα σκορ που θα οδηγηθεί ο κάθε παίχτης κάνοντας την συγκεκριμένη κίνηση, τότε αν μπορούσαμε να εξασφαλίσουμε με κάποιο τρόπο ότι ο min θα παίξει sub-optimal, και η ρίζα επέλεγε το 1 έναντι του 3, τότε, ο MAX θα μπορούσε να έχει path νίκης σε δύο κινήσεις, έναντι της βέλτιστης επιλογής που έκανε, δεδομένου ότι δεν ξέρουμε τις κινήσεις του MIN, εξασφαλίζοντας μόνο ότι δεν θα χάσει και θα κάνει σίγουρες κινήσεις υπέρ του.

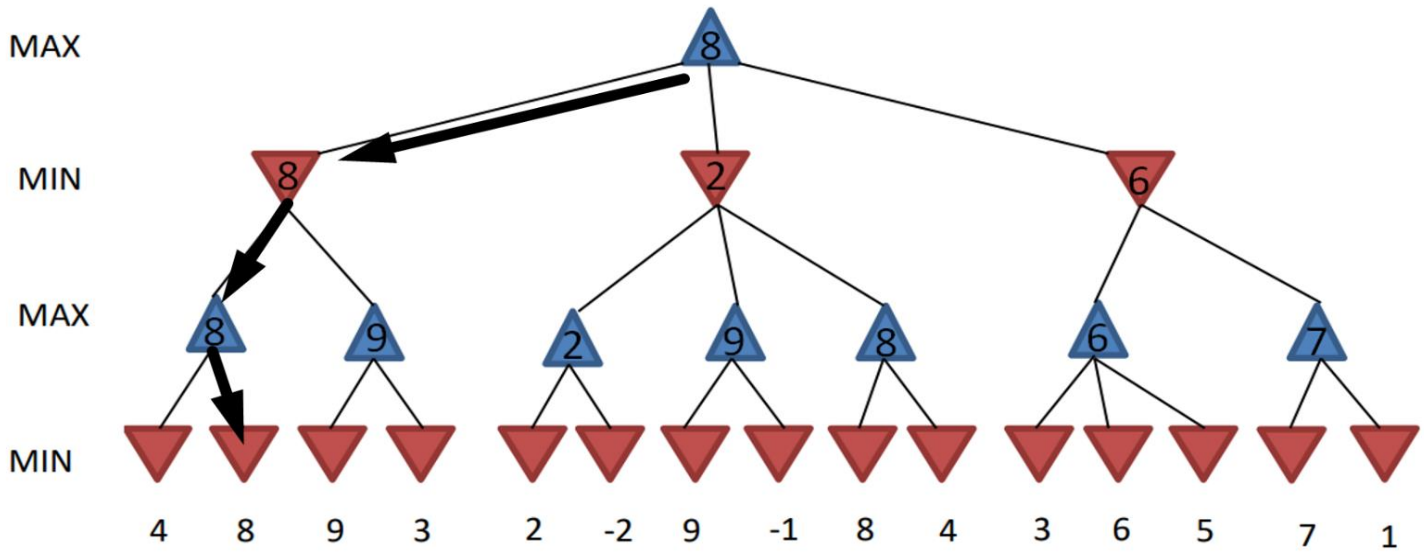


Πρόβλημα 2

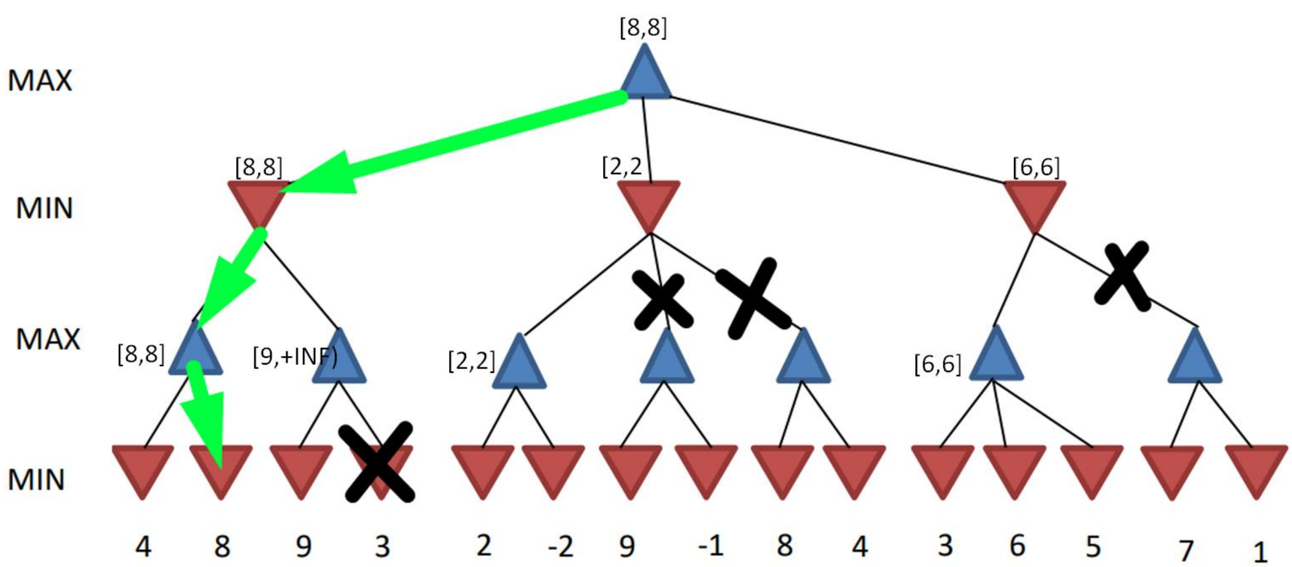
(α) Για κάθε κόμβο που δεν είναι φύλλο, να υπολογίσετε την minimax τιμή του.



(β) Ποια είναι η minimax απόφαση στη ρίζα του δένδρου;



(γ) Να δώσετε όλους τους κόμβους οι οποίοι κλαδεύονται από τον αλγόριθμο ALPHA-BETA-SEARCH όταν αυτός εκτελεστεί για το παραπάνω δένδρο. Να υποθέσετε ότι τα παιδιά ενός κόμβου επισκέπτονται από τα αριστερά προς τα δεξιά



Στην παραπάνω εικόνα βλέπουμε με X τους κόμβους οι οποίοι δεν παίρνονται ποτέ υπόψιν. Για τον κόμβο MIN με value = 3, δεν χρειάζεται να ποτέ τον ελέγξουμε από τη στιγμή που το δίπλα του node είχε value = 9 > 8, αφού ο κόμβος MAX θα πάρει τιμές από [9, +INF), και παραπάνω ο κόμβος MIN δεν θα τον επιλέξει ποτέ, από τη στιγμή που υπάρχει το [8,8].

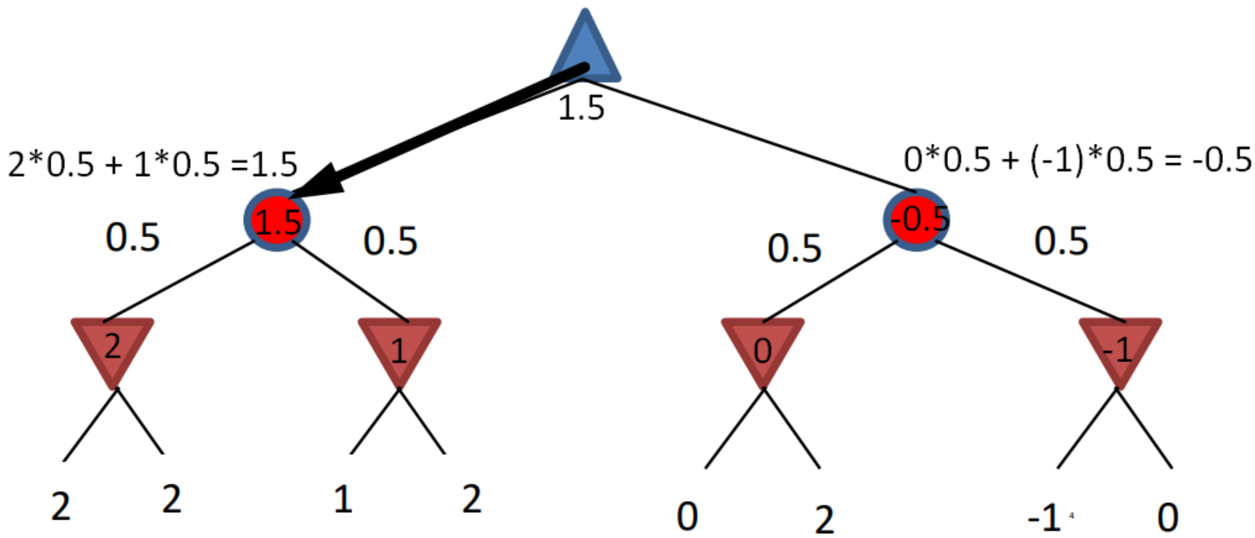
Για τους μεσαίους κόμβους που κλαδέψαμε, αφού ο κόμβος MIN [2,2] είναι μικρότερος του [8,8] ότι τιμή και να είχαν οι υπόλοιποι κόμβοι ποτέ δεν θα προτιμηθούν από την ρίζα.

Για το δεξιά ακριανό κλάδεμα, αφού πάλι ο αριστερά κόμβος πήρε τιμή [6,6] < [8,8] τότε ο MIN δεν θα τον ξεπεράσει ποτέ ώστε μετά να επιλεχτεί από την ρίζα.

Να σημειωθεί πως η ρίζα ελέγχει κατά βάθος και από αριστερά προς τα δεξιά.

Πρόβλημα 3

(α) Να αντιγράψετε το δένδρο, να υπολογίσετε τις τιμές των εσωτερικών κόμβων και να δείξετε με ένα βελάκι ποια είναι η καλύτερη κίνηση για την ρίζα.



(β1) Αν μας δώσουν τις τιμές των πρώτων έξι φύλλων όπως φαίνονται στο παραπάνω σχήμα, χρειάζεται να υπολογίσουμε τις τιμές του έβδομου και του όγδοου κατά την εύρεση της καλύτερης κίνησης για τη ρίζα;

Ας υποθέσουμε ότι έχουμε τις τιμές των πρώτων έξι φύλλων εκτός των δύο τελευταίων. Τότε η συνάρτηση του δεξιά κόμβου τύχης, θα είναι της μορφής: 0*0.5 + χ*0.5

Αυτό σημαίνει πως η τιμή του χ θα προκύψει από την επιλογή του MIN κόμβου. Αν το χ που θα βρεθεί ας πούμε έχει την τιμή 10, τότε έχουμε 0.5*10 = 5 > 1.5 , οπότε ο MAX κόμβος θα οδηγηθεί σε μη βέλτιστη λύση και έτσι ο ισχυρισμός μας καταρρίπτεται.

(β2)Αν μας δώσουν τις τιμές των πρώτων επτά φύλλων όπως φαίνονται στο παραπάνω σχήμα, χρειάζεται να υπολογίσουμε τις τιμές του όγδοου; Να λάβετε υπόψη σας ότι οι δυνατές τιμές για τα φύλλα είναι από $-\infty$ έως $+\infty$.

Αν μας δώσουν τις τιμές των πρώτων εφτά φύλλων, τότε οι τιμές του δεξιότερου min θα ανήκουν στο διάστημα από (-INF,-1], χωρίς καν να εξετάσουμε άλλες τιμές. Αυτό ισχύει γιατί ο MIN κόμβος μπορεί μόνο να επιλέξει κάτι χειρότερο του -1, και με βάση τις υπόλοιπες τιμές του παραπάνω σχήματος, τότε ότι τιμή και να είχαμε για $8^{η}$, η ρίζα πάντα θα επέλεγε το αριστερά path. Όσο μικρότερη είναι η τιμή του MIN, τόσο χειρότερη είναι και η συνεισφορά της στον κόμβο τύχης.

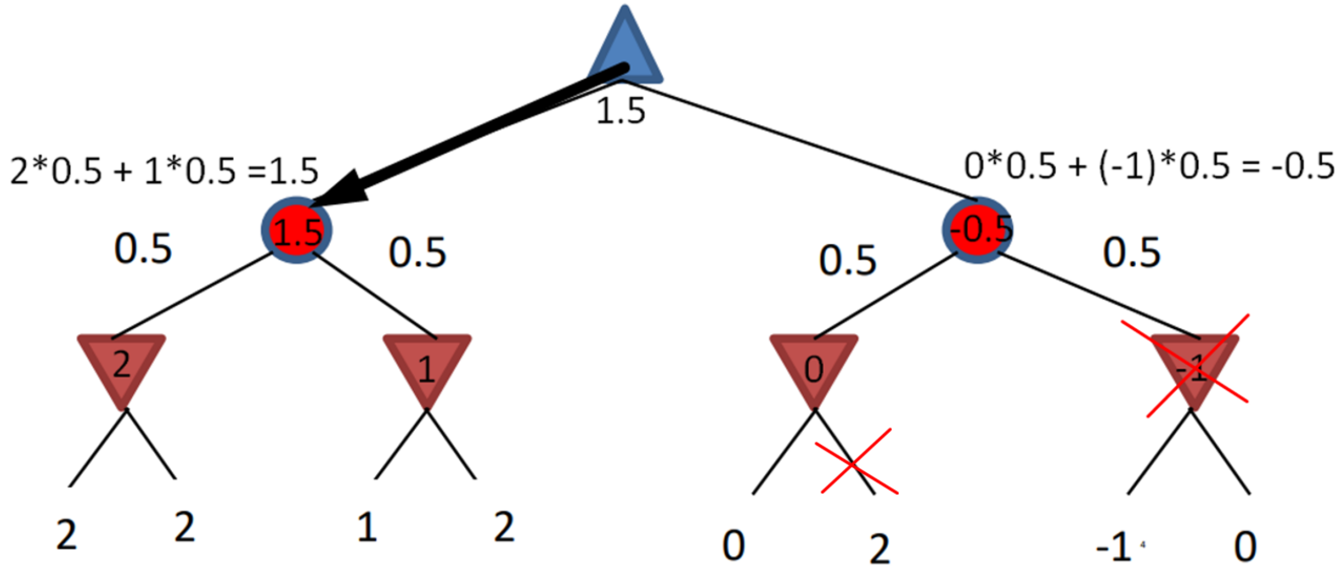
(γ) Υποθέστε ότι οι τιμές των φύλλων βρίσκονται στο διάστημα [−2,2]. Αφού αποτιμηθούν τα δύο πρώτα φύλλα, ποιες είναι οι δυνατές τιμές του αριστερού κόμβου τύχης;

Ας υποθέσουμε ότι ο αριστερά κόμβος MIN επιλέξει την τιμή MIN1, και ο δεξιά του επιλέξει την τιμή MIN2. Τότε από τον κόμβο τύχης θα έχουμε την παρακάτω εξίσωση.

0.5*(MIN1 Ε[-2,2]) + 0.5*(MIN2 Ε[-2,2]) = MIN1 Ε[-1,1] + MIN2 Ε[-1,1] = result Ε[-2,2]

Αφού τα MIN1 και MIN2 θα κυμαίνονται μεταξύ -1 και 1 για τον κόμβο τύχης, τότε το τελικό result θα μπορεί να έχει τιμές από -2 εώς και 2 οι οποίες θα είναι και οι ακριανές τιμές που θα μπορεί να έχει.

(δ) Υποθέστε ότι τρέχουμε ένα αλγόριθμο όπως το κλάδεμα άλφα-βήτα για το παραπάνω δένδρο και επιπλέον γνωρίζουμε ότι οι τιμές των φύλλων είναι στο διάστημα [−2,2]. Σημειώστε με ένα κυκλάκι τους κόμβους που δεν χρειάζεται να αποτιμηθούν κατά τον υπολογισμό της καλύτερης κίνησης για τη ρίζα.



Για το παραπάνω πρόβλημα, παίρνοντας υπόψιν ότι έχουμε δύο κόμβους τύχης, και το διάστημα τιμών των φύλλων είναι μεταξύ [-2,2] τότε ο κόμβος που δεν θα χρειαστεί να αποτιμηθεί, είναι ο δεξιότερος. Αυτό ισχύει γιατί ο αριστερά κόμβος τύχης έφερε αποτέλεσμα 1.5, και από τον δεξιά κόμβο, ο αριστερός του min είχε μηδενική συνεισφορά στην τιμή του κόμβου τύχης. Όπως αποδείξαμε και στο ερώτημα γ, οι τιμές για ένα MIN κόμβο, όταν υπολογίζουμε το άθροισμα κόμβου τύχης, μπορούν να πάρουν τιμές από [-1,1] που σε κάθε περίπτωση είναι μικρότερο του 1,5.
Έτσι, έχοντας τιμές φύλλων στο διάστημα [-2,2] ότι τιμές και να είχε ο δεξιά κόμβος MIN του δεξιού κόμβου τύχης, ποτέ δεν θα μπορούσε να ξεπεράσει το 1,5 και να προτιμηθεί από την MAX ρίζα.
0*0.5 + Χε[-2,2]*0.5 = 0 + Χε[-1,1] < 1,5

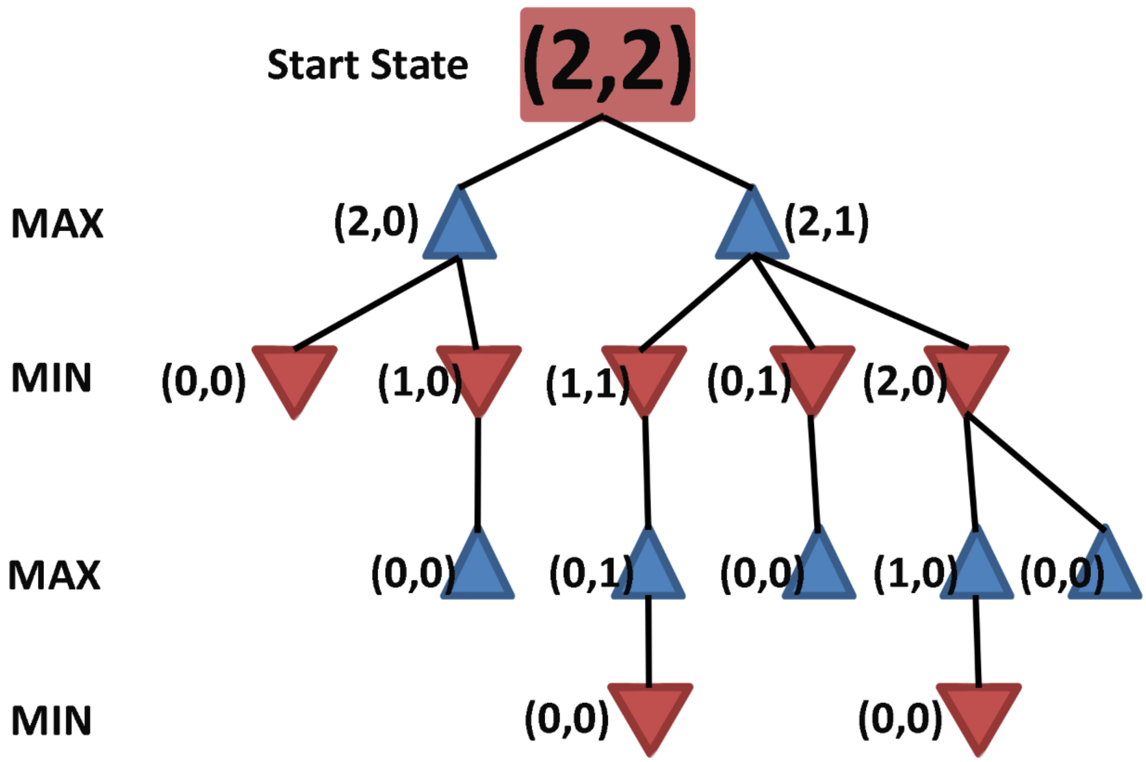
Για τον δεξιά κόμβο MIN με τιμές 0 και 2, το δύο θα αποκοπεί από τον κόμβο για αντίστοιχο λόγο. Εφόσον βρήκαμε το 0, ο κόμβος MIN δεν θα προτιμήσει ποτέ κάτι καλύτερο του, οπότε οι τιμές που μπορεί να πάρει ο κόμβος MIN θα είναι στο διάστημα [-2,0] που όποια τιμή και να πάρουμε δεν θα έχουμε ποτέ συνολικό αποτέλεσμα κόμβου τύχης μεγαλύτερο του 1,5 που βρέθηκε από τον αριστερά κόμβο τύχης. Έτσι βλέπουμε στο παραπάνω σχήμα ποιοι κόμβοι θα αποκοπούν.

Το πρόβλημα 4 ακολουθεί παρακάτω

Πρόβλημα 4 Nim game 2-2 version

Θα θεωρήσουμε την έκδοση του παιχνιδιού με 2 στοίβες 2 όμοιων αντικειμένων και θέλουμε να κερδίζει το παιχνίδι ο παίκτης που θα αφαιρέσει το τελευταίο εναπομένον αντικείμενο. Έχετε να κάνετε τα εξής:

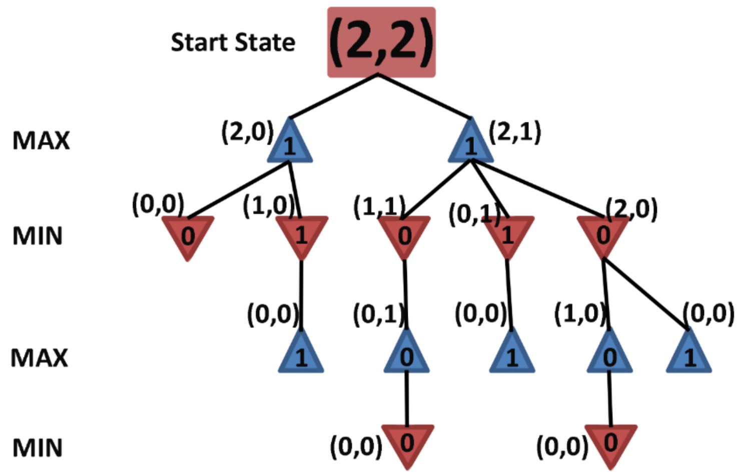
1. Σχεδιάστε το πλήρες δένδρο παιχνιδιού υποθέτοντας ότι ο παίκτης MAX παίζει πρώτος.



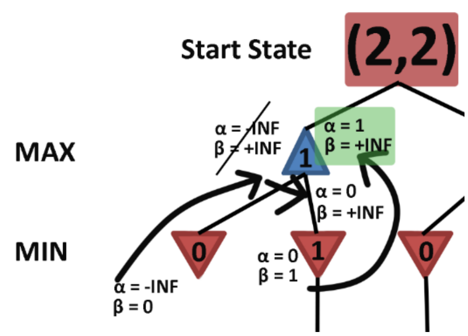
Στην παραπάνω εικόνα βλέπουμε το δέντρο παιχνιδιού αν έπαιζε ο MAX πρώτος, τα (X,X) συμβολίζουν το πόσα σπίρτα μένουν σε κάθε πλευρά του pile. Όλα όσα έχουν (0,0) συμβολίζουν τα win/lose moves υπέρ η κατά του max.

2. Θεωρήστε ότι η συνάρτηση χρησιμότητας στα φύλλα του δένδρου παιχνιδιού είναι 0 για τις καταστάσεις που κερδίζει ο MIN και +1 για τις καταστάσεις που κερδίζει ο MAX. Εκτελέστε τον αλγόριθμο κλαδέματος άλφα-βήτα (ALPHA-BETA-SEARCH στις διαφάνειες) για να υπολογίσετε τη minimax τιμή στη ρίζα του δένδρου. Θα πρέπει να δείξετε και τις τιμές των α και β καθώς ο αλγόριθμος εξερευνά το δένδρο. Ποιο τμήμα του δένδρου από το ερώτημα 1, κλαδεύεται από τον αλγόριθμο;

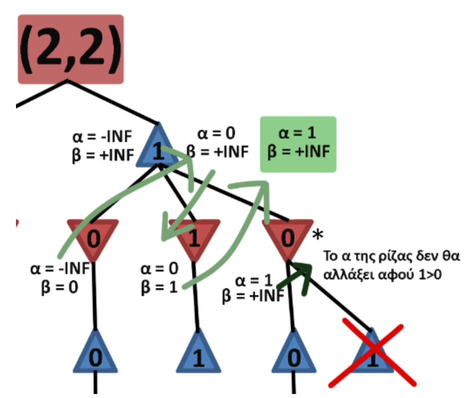
Θα τρέξουμε τον αλγόριθμο alpha-beta για τα 2 game paths που υπάρχουν τα οποία είναι το (2,0) και το (2,1).



<u>Για το (2,0)</u>: Αρχικά θα έχουμε α = -INF και β = +INF. Στην αρχή θα πάμε κάτω αριστερά στον κόμβο MIN και έπειτα θα αλλάξουμε την β = 0. Οπότε οι τιμές α και β στην ρίζα θα γίνουν α = 0 και β=+INF, και θα περάσουμε στον δεξιά MIN κόμβο. Από εκεί θα πάρει την τιμή β = 1, και θα επιστρέψει στην ρίζα τις τιμές που βρήκε, οι οποίες θα πάνε ως εξής. Θα έχουμε α = 1 και β = +INF.



<u>Για το (2,1)</u>: Αρχικά πάλι θα έχουμε α = -INF και β = +INF. Το σκεπτικό για τις τιμές α και β, είναι το ίδιο με την εκτέλεση του (2,0) game tree. Τώρα, ο κόμβος που έχει σημαδευτεί με το κόκκινο X, θα αποκοπεί από το δέντρο, αφού πήραμε σαν value που προορίζεται για MIN node το 0, τότε αυτό δεν πρόκειται ποτέ να επιλέξει κάτι μεγαλύτερο από αυτό. Εμείς όμως έχουμε βρει ήδη το 1 σαν επιλογή για τον MAX οπότε το τελικό α της ρίζας που θα είναι και τιμή της, δεν θα υπολογιστεί από τον αριστερά και τον δεξιά κόμβο από την ρίζα αλλά από τον μεσαίο.



Σημείωση: Επειδή μας νοιάζει ο MAX παίχτης τι θα κάνει θεώρησα ότι τα φύλλα είναι αυτά που φαίνονται στις φωτογραφίες και έχουν μέσα μόνο values , και δεν αποτελούν μέρος της εκτέλεσης του minimax. Αυτό το έκανα γιατί μου φάνηκε αρκετά περίπλοκο να τεκμηριώσω σε χαρτί και να σχεδιάσω όλη την εκτέλεση του alpha-beta pruning, παρόλο που έχω κατανοήσει όλο το σκεπτικό του, όπως είδατε παραπάνω.

3. Αν και οι δύο παίκτες παίζουν αλάνθαστα, ποιος θα κερδίσει; Εξηγήστε την απάντηση σας χρησιμοποιώντας την ορολογία του μαθήματος.

Από την αρχή κιόλας του παιχνιδιού, ο MAX είναι καταδικασμένος να χάσει ενάντια σε έναν αλάνθαστο αντίπαλο. Αυτό συμβαίνει γιατί όποιο αρχικό game path και να ακολουθήσει, αν ο παίκτης MIN παίξει με τον βέλτιστο τρόπο, ο οποίος θα είναι η κίνηση κόστους ίσο με 0 (που σημαίνει ήττα του MAX) τότε ο MAX όποια επιλογή και να ακολουθήσει από εκεί και ύστερα, δεν έχει περιθώριο για νίκη. Με άλλα λόγια, αν ο κάθε παίκτης έτρεχε ξεχωριστά τον minimax αλγόριθμο, από πλευράς του, τότε πάντα ο πρώτος παίκτης θα έχανε. Ο καθένας θα επέλεγε την μεγαλύτερου κόστους λύση δηλαδή την 1, όπου υπήρχε η δυνατότητα στο κάθε game state. Έτσι, το τελικό συμπέρασμα είναι η νίκη είναι πολωμένη προς 2^{ο} παίκτη. Αν ο MIN δεν έπαιζε αλάνθαστα, τότε το καλύτερο game path για τον παίκτη 1, είναι αρχικά να τραβήξει ένα σπίρτο, καθώς δίνει στον παίκτη 2, μεγαλύτερο περιθώριο λάθους, έτσι ώστε ο 1^{ος} να κερδίσει το παιχνίδι.

Σας ευχαριστώ πολύ για τον χρόνο σας, καλή διόρθωση 😊