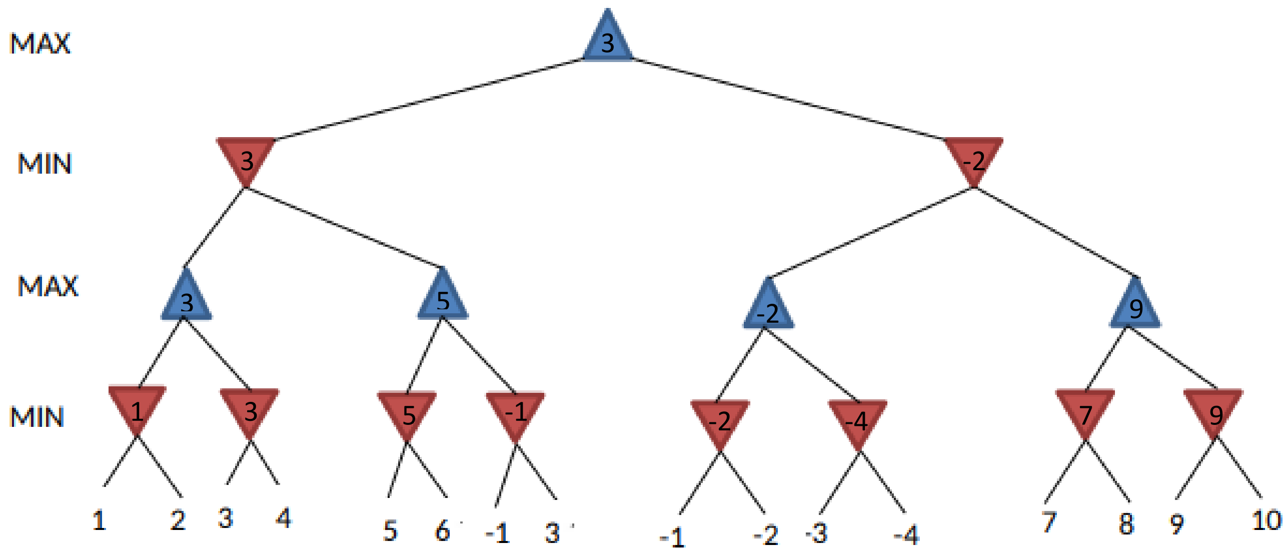
Τεχνητή Νοημοσύνη 1 – Χειμερινό 2021-2022

Εργασία δεύτερη

Κατσαούνη Σοφία Μερόπη , sdi1800070

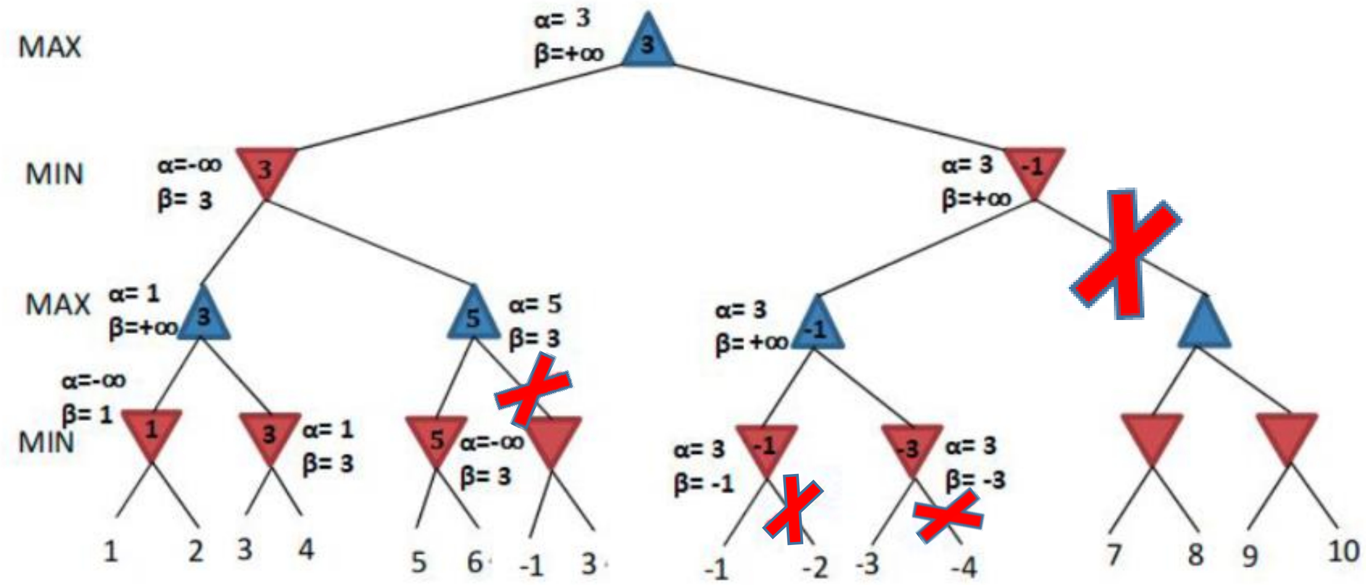
Πρόβλημα 3:

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



(β) Με βάση το παραπάνω ερώτημα , η minimax απόφαση έτσι όπως βλέπουμε το δένδρο θα είναι ο MAX να κατέβει στο αριστερό παιδί του, και εκεί με την σειρά του ο MIN να κατέβει παλι στο αριστερό του παιδί. Στη συνέχεια ο MAX θα κατέβει στο δεξί του παιδί επιτυγχάνοντας έτσι την μέγιστη αξία για αυτόν στο δεδομένο δέντρο με τις δεδομένες συνθήκες.

(γ) Χρησιμοποιώντας τον αλγόριθμο ALPHA-BETA-SEARCH, παρατηρούμε ότι υπάρχει βελτίωση στις επιδώσεις του αλγορίθμου minimax καθώς έξυπνα παραλείπουμε κάποιους κόμβους μαζί με τα παιδία τους περιορίζοντας έτσι σημαντικά το συνολικό πλήθος των κόμβων που χρειάζεται να επισκεφτούμε. Η λογική είναι ότι ο MAX θα επιλέγει πάντα την μέγιστη επιλογή και ο MIN θα επιλέγει πάντα την ελάχιστη επιλογή καθώς και θα γίνεται χρήση μιάς μορφής ’’άνω και κάτω φραγμάτων'' για να προσδιορίζεται εάν ένας κόμβος πρέπει να κλαδευτεί ή όχι.

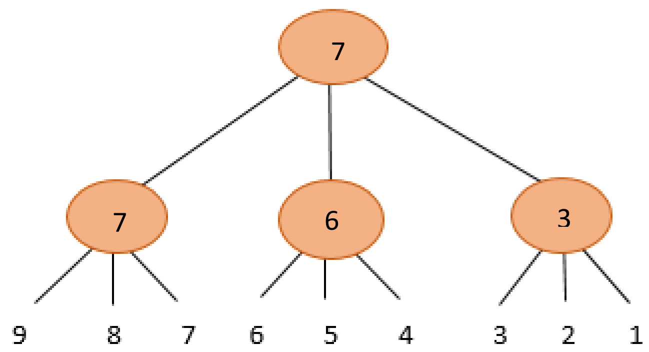


Με αρχικές τιμές των α και β ως α=-∞ και β=+∞ ο αλγόριθμος εξετάζει πρώτα τους αριστερότερους κόμβους. Φτάνοντας στις τερματικές καταστάσεις 1 και 2, το β ανανεώνεται με β=1 εφόσον είναι η μικρότερη τιμή. Ο αλγόριθμος επιστρέφει την τιμή 1 στον εξεταζόμενο κόμβο (αριστερότερος κατω κόμβος, ή MIN 1). Προχωρώντας προς τα πάνω σε βάθος 2 η τιμή του α ανανεώνεται σε 1 εφόσον 1>-∞. Ο MAX κόμβος στην συνέχεια περνάει τις τιμές του στον δεξί MIN παιδί κόμβο οπου το β ανανεώνεται σε β=3 εφόσον 3<+∞. Ο MIN κόμβος αποκτά την μικρότερη τιμή 3 και έτσι ο MAX κόμβος ανανεώνεται με τιμή 3. Επίσης το β στον MIN κόμβο του βάθους 1 ανανεώνεται σε β=3. Ο MIN κόμβος βάθους 1 περνάει τις τιμές α=-∞ και β=3 στο δεξί MAX παιδί και αυτό με την σειρά του περνάει τις τιμές στο αριστερό MIN παιδί με τερματικές καταστάσεις 5 και 6. Ο αλγόριθμος επιστρέφει την τιμή 5 σε αυτόν τον κόμβο και η τιμή του α στον MAX βάθους 2 γίνεται 5. Στο σημείο αυτό ο αλγόριθμος βρίσκεται στον MAX κόμβο και συγκρίνει αν η τιμή του γείτονα είναι μεγαλύτερη του β δηλαδή αν 3>=3, αληθές αρα επιστρέφει την τιμή 5 στον κόμβο και κλαδεύεται ο άλλος γειτονικός MIN κόμβος. Στον MIN κόμβο βάθους 1 επιλέγεται η τιμή 3 από αυτήν του 5. Η τιμή α στην ρίζα αποκτά τιμή α=3.

Στην συνέχεια οι τιμές α=3 και β=+∞ περνούν από την ρίζα μέχρι και τον κόμβο με τις τερματικές καταστάσεις -1 και -2. Ο αλγόριθμος συγκρίνει την τιμή του γείτονα με την τιμή του α δηλαδή αν -1<=3 αληθές άρα επιστρέφεται η τιμή -1 στον MIN κόμβο βάθους 3 και κλαδεύεται ο γείτονας του. Γυρνάμε πίσω στον MAX και ελέγχουμε το άλλο παιδί MIN κόμβο αυτό με τις τερματικές καταστάσεις 3 και 4. Η τιμή του β γίνεται -3, και ο αλγόριθμος συγκρίνει την τιμή του γείτονα με τον α δηλαδή αν -3<=3 αληθές άρα επιστρέφεται η τιμή -3 στον MIN κόμβο και κλαδεύουμε τον άλλον γείτονα του. Στον MAX επιπέδου 2 επιστρέφεται η τιμή -1. Στον MIN κόμβο επιπέδου 1 ο αλγόριθμος συγκρίνει την τιμή του γείτονα με την τιμή του α δηλαδή -1<=3 αληθές άρα κλαδεύεται ο άλλος γειτονικός MAX κόμβος. Τελικά επιστρέφεται η τιμή 3 στην ρίζα.

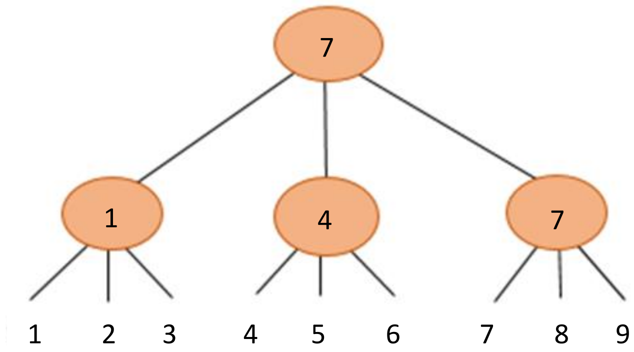
Πρόβλημα 2:

Παρατηρώ ότι οταν οι τιμές στα φύλλα του δέντρου να είναι σε φθίνουσα ταξινόμηση, έχουμε μέγιστο κλάδεμα κόμβων στο δέντρο.



Αφού παραχθούν οι πρώτες 3 τερματικές καταστάσεις, ο α θα έχει την τιμη 7. Ο επόμενος Min κόμβος θα εξετάσει εάν 6<=7 αληθες, αρα οι υπόλοιποι κόμβοι φεύγουν. Ο επόμενος Min κόμβος θα εξετάσει εάν 3<=6 αληθές αρα οι υπόλοιποι κόμβοι θα κλαδευτούν. Έτσι θα έχουμε κλαδέψει τον μέγιστο αριθμό κόμβων. Σε μια γενίκευση αυτού του παραδείγματος η ιδιότητα που πρέπει να έχουν οι τιμές χρησιμότητας στα φύλλα του δένδρου ώστε ο αριθμός των κόμβων που κλαδεύεται με την τεχνική άλφα-βήτα να είναι μέγιστος είναι η MIN επιλογή του πρώτου παιδιού να είναι μεγαλύτερο νούμερο σε σχέση με τα πρώτα φύλλα των υπόλοιπων παιδιών του δέντρου.(Π.χ εδώ πράγματι το 7>6 και 7>3)

Αντίστοιχα, η ιδιότητα που πρέπει να έχουν οι τιμές χρησιμότητας στα φύλλα του δένδρου ώστε ο αριθμός των κόμβων που κλαδεύεται με την τεχνική άλφα-βήτα να είναι ελάχιστος, είναι οι τιμές στα φύλλα του δέντρου να είναι σε αύξουσα ταξινόμηση.



Αφού παραχθούν οι πρώτες 3 τερματικές καταστάσεις, ο α θα έχει την τιμη 1. Ο επόμενος Min κόμβος θα εξετάσει εάν 4<=1 ψευδές, αρα οι κόμβοι παραμένουν. Ο επόμενος Min κόμβος θα εξετάσει εάν 7<=4 ψευδές, αρα οι κόμβοι παραμένουν.

Πρόβλημα 4:

α) Στο max δέντρο υπάρχουν μόνο MAX κόμβοι και στον αλγόριθμο alpha-beta γίνεται η σύγκριση εάν ένας κόμβος είναι μεγαλύτερος του b, το b όμως θα παραμείνει σταθερά +∞ εφόσον δεν υπάρχουν MIN κόμβοι με αποτέλεσμα να μην μπορούν να κλαδευτούν κόμβοι και άρα ο αλγόριθμος να αχρηστευτεί.

β) Οι κόμβοι chance έχουν τυχαίες τιμές με αποτέλεσμα να μην υπάρχει η έννοια της βέλτιστης κίνησης για τον αντίπαλο του Max. Αυτό θα έχει σαν αποτέλεσμα η τιμή του b να μην μπορεί να οριστεί γιατί το περιεχόμενο των μη εξερευνημένων παιδιών μπορεί να φέρει τεράστιες αλλαγές στις τιμές των chance κόμβων και έτσι δεν μπορούμε να κλαδέψουμε κανέναν κόμβο. Ετσι και εδώ ο αλγόριθμος alpha-beta δεν είναι ιδιαίτερα χρήσιμος.

γ) Στο δέντρο max υπάρχουν μόνο κόμβοι ενός είδους (τύπου MAX). Αυτό έχει σαν αποτέλεσμα να μην αλλάξει ποτέ η τιμή του b και άρα δεν θα υπάρξει και ποτέ κλάδεμα κόμβου.

δ) Οι κόμβοι chance έχουν τυχαίες τιμές με αποτέλεσμα να μην υπάρχει η έννοια της βέλτιστης κίνησης για τον αντίπαλο του Max. Αυτό θα έχει σαν αποτέλεσμα η τιμή του b να μην μπορεί να οριστεί γιατί το περιεχόμενο των μη εξερευνημένων παιδιών μπορεί να φέρει τεράστιες αλλαγές στις τιμές των chance κόμβων και έτσι δεν μπορούμε να κλαδέψουμε κανέναν κόμβο.

ε) Εάν οι τιμές είναι θετικές ή μηδέν δεν επηρεάζει επομένως ισχύουν τα ίδια με το γ.

στ) Εάν οι τιμές είναι θετικές ή μηδέν δεν επηρεάζει επομένως ισχύουν τα ίδια με το δ.

ζ) Όπως είπα στο ερώτημα ε μηδενικές ή θετικές τιμές δεν έχουν διαφορετική επίπτωση στο εάν θα κλαδευτεί κάποιος κόμβος.

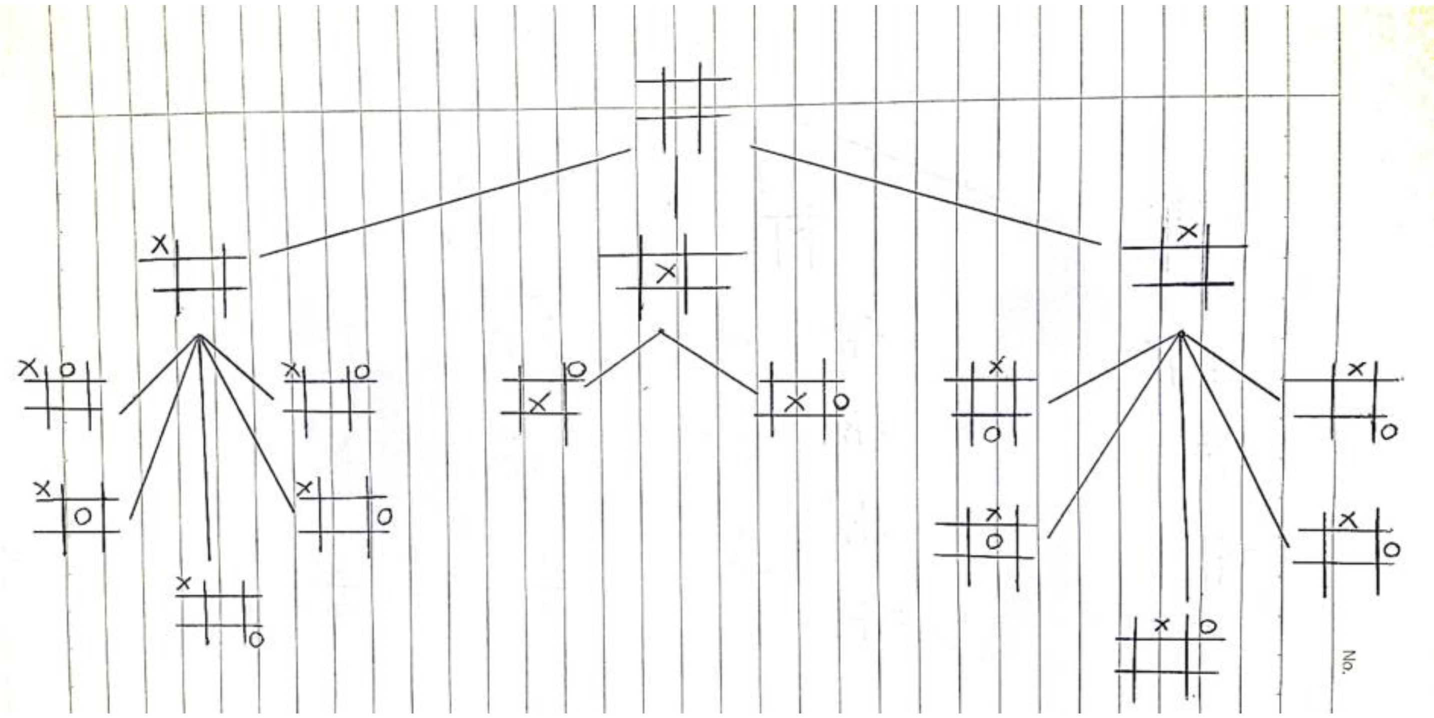
η) Όπως είπα στο ερώτημα στ μηδενικές ή θετικές τιμές δεν έχουν διαφορετική επίπτωση στο εάν θα κλαδευτεί κάποιος κόμβος.

Πρόβλημα 1:

1) 9! = ο αριθμός των ακολουθιών κινήσεων που γεμίζουν το ταμπλό. Υπάρχουν 9 κουτάκια και άρα 9 πιθανές κινήσεις για τον πρώτο παίκτη αντίστοιχα μετα 8 πιθανές κινήσεις για τον δεύτερο παίκτη κλπ.

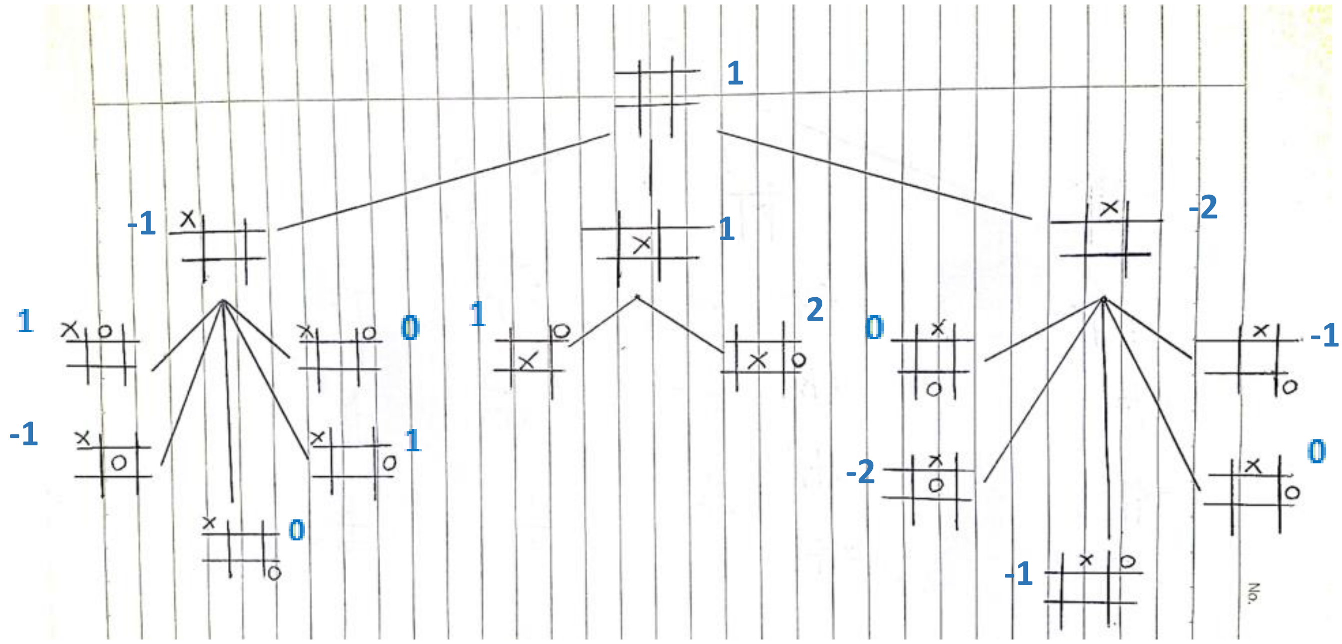
2)

Από το δέντρο έχουν αφαιρεθεί οι συμμετρικές καταστάσεις



(Το βάθος του δέντρου είναι 2 απλά το σχεδίασα με τέτοια τρόπο για να είναι ίσως πιο καθαρό)

3) Από την εκφώνηση δίνεται ότι $Eval(s) = 3X_2(s) + X_1(s) – (3O_2(s) + O_1(s))$ Επομένως μετρώντας τον αριθμό στηλων γραμμών και διαγωνιών που βρίσκονται τα Χ και Ο σε κάθε τρίλιζα προκύπτει:

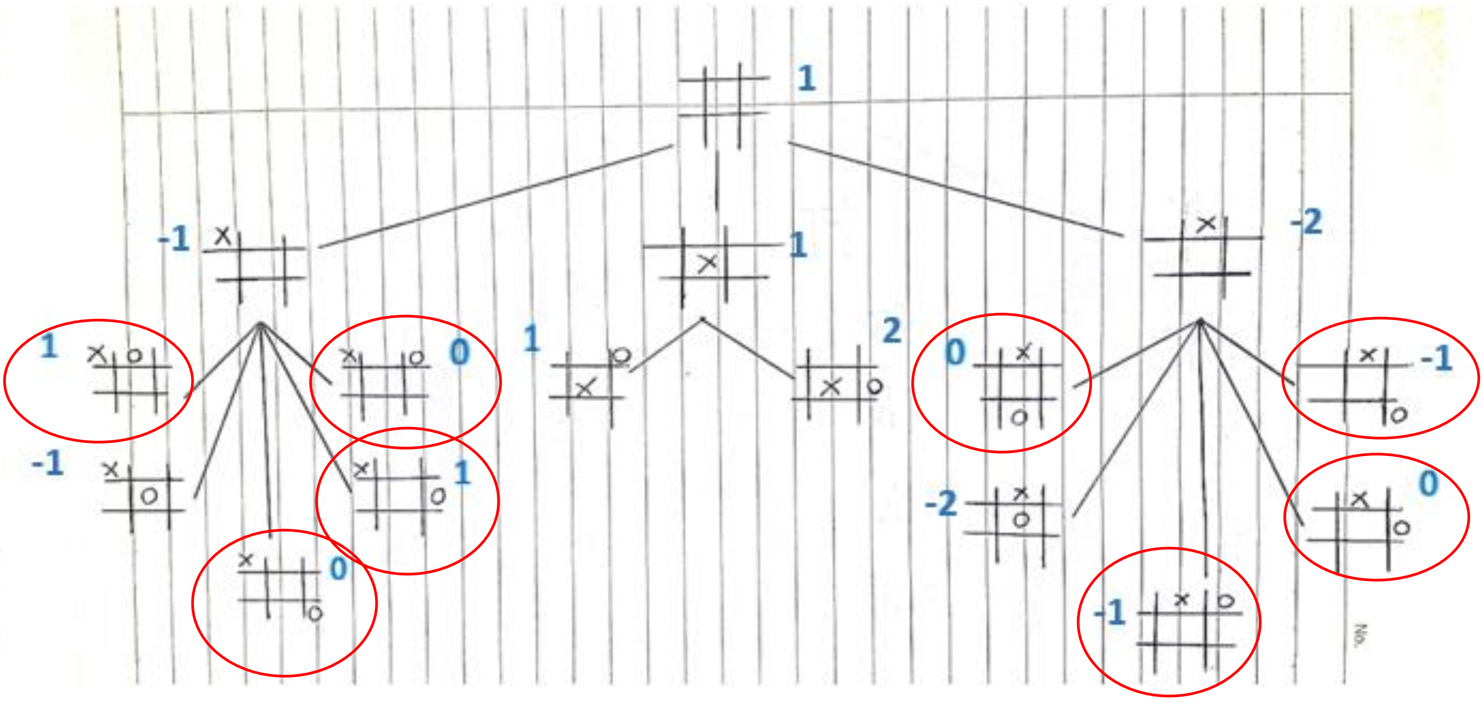


4)

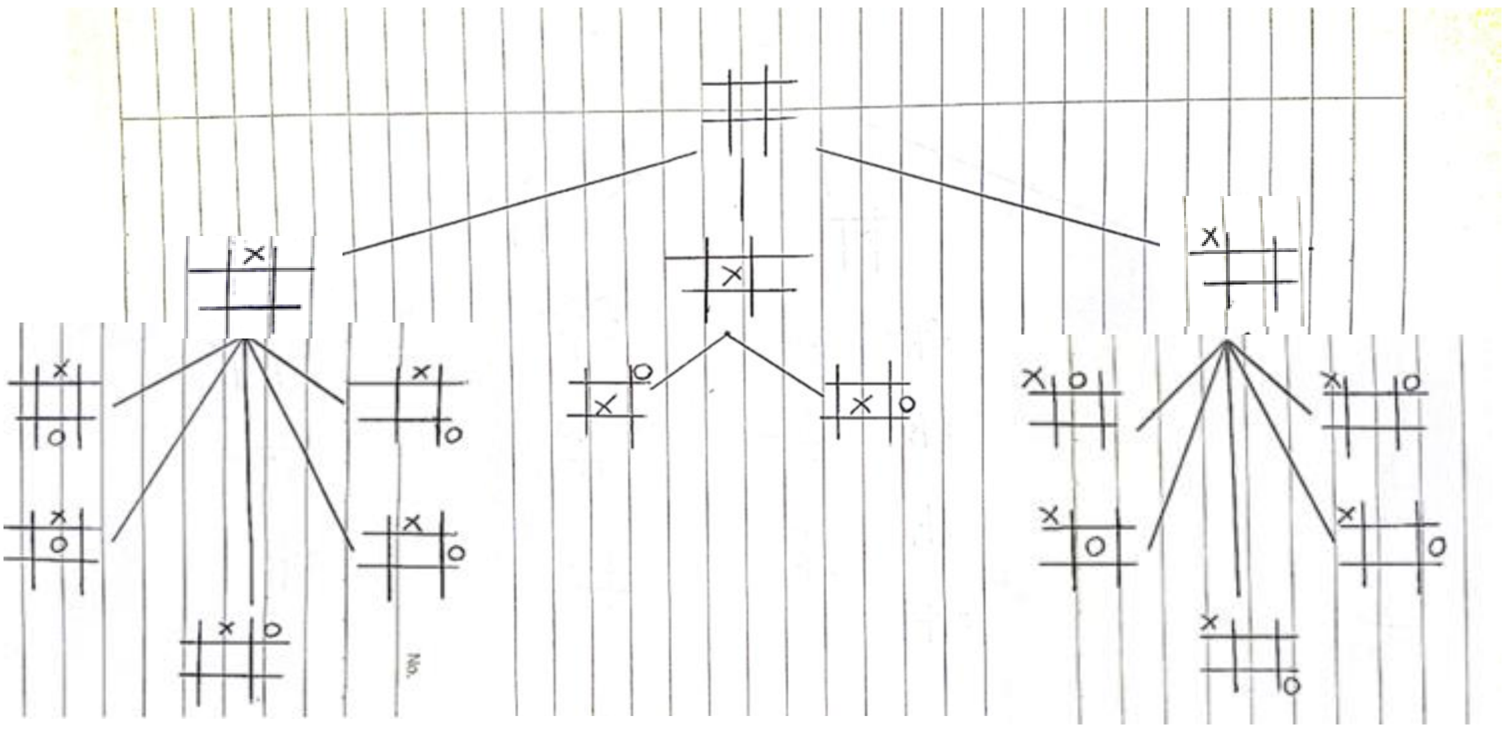
Η minimax απόφαση στη ρίζα του δέντρου θα είναι 1. Εκτελώντας τον αλγόριθμο, αναδρομικά κατεβαίνουμε όλη τη διαδρομή μέχρι τα φύλλα του δένδρου παιχνιδιού και μετά οι τιμές minimax αντιγράφονται προς τα πίσω μέσω του δένδρου όπως εκτυλίσσεται ή αναδρομή. Στο αριστερό παιδί η συνάρτηση χρησιμότητας επιστρέφει τιμή -1 εφόσον το παιδί κρατάει το MIN value. Στο μεσαίο παιδί επιστρέφεται η τιμή 1 και στο δεξιό παιδί επιστρέφεται η τιμή -2 για τον ίδιο λόγο. Εφόσον στην ρίζα του δέντρου έχουμε MAX επιλογή επιλέγεται το 1 και επιστρέφεται. Επομένως καταλήγουμε ότι η βέλτιστη κίνηση στη ρίζα του δεντρου θα είναι να μπεί το Χ στο κέντρικο κουτάκι της τρίλιζας.

5)

Παρατηρούμε ότι το MIN-MAX value στη ρίζα επηρεάζεται από τις τιμές -1, 1, 2 και -2. Εκτελώντας τον αλγόριθμο ALPHA-BETA-SEARCH των διαφανειών βλέπουμε ότι οι υπόλοιποι κόμβοι εκτός των προαναφερθέντων θα κλαδευτούν εφόσον δεν έχουν ιδιαίτερη επιρροή στην επιλογή της βέλτιστης κίνησης για κάθε επίπεδο.



Αν οι κόμβοι παράγονται με την αντίστροφη σειρά το δέντρο θα ήταν της μορφής:



Οπώς παρατηρούμε δεν υπάρχει ιδιαίτερη βελτίωση απόδοσης στον αλγόριθμο ALPHA-BETA-SEARCH εφόσον θα κλαδευτεί η ίδια ποσότητα κόμβων. Η βέλτιστη σειρά παραγωγής κόμβων είναι αυτήν που φαίνεται και παραπάνω.