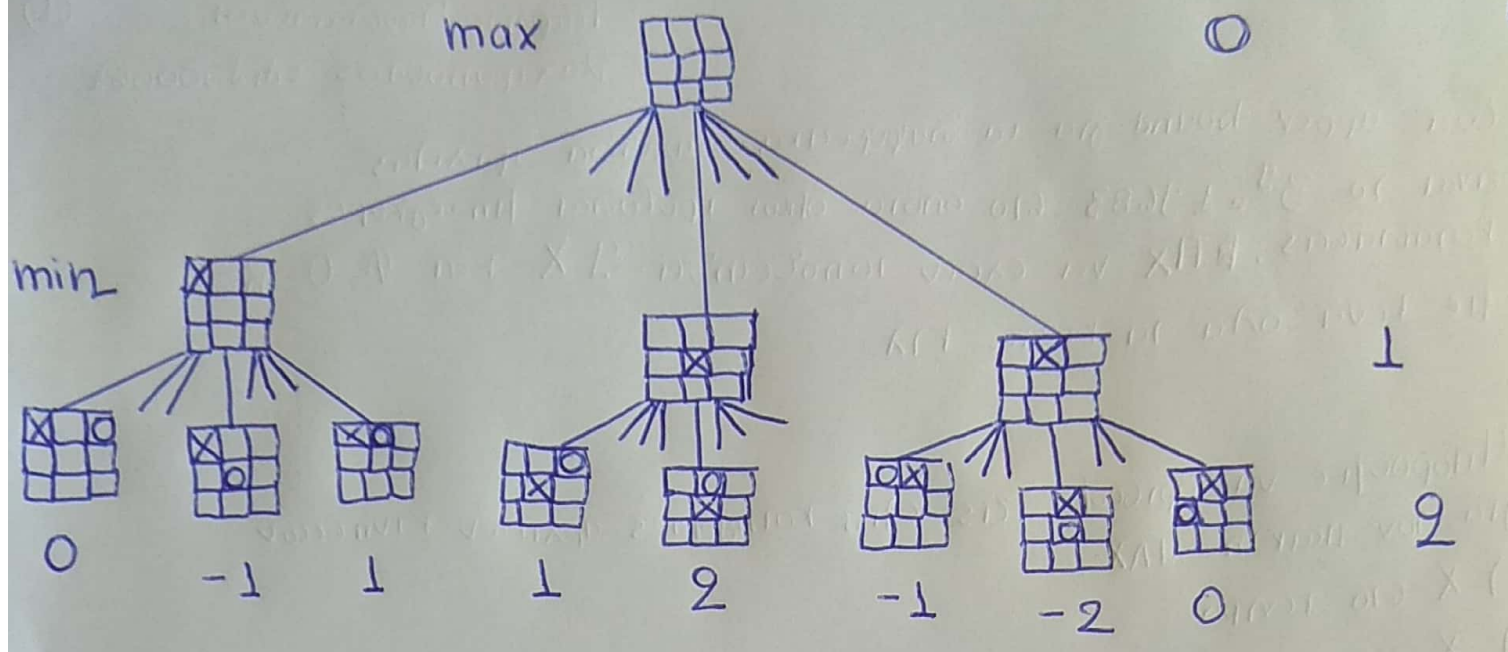


1. Ένα upper bound για τα διαφορετικά παιχνίδια τριλίας είναι το $3^9 = 19683$ στο οποίο όμως κρίνονται μη-εγκυρες καταστάσεις. Πχ να έχουν τοποθετηθεί 9 X και 0 0 ή με κενά όλα τα κελιά κτλ.
2. Μπορούμε να ορίσουμε τις εής κατηγορίες αρχικών κινήσεων για τον παίκτη MAX.
 - i) X στο κενό
 - ii) X σε γωνία
 - iii) X ανάμεσα σε γωνίεςοπότε μπορούμε να παραβλέψουμε τις βελτιστοποιη κινήσεις στο βήμα. Επίσης μπορούμε να ορίσουμε αντίστοιχες ^{κατηγορίες} αρχικών κινήσεων του παίκτη MIN.
 - i) Αν υπάρχει X σε κάποια γωνία τότε για 0 στις υπόλοιπες διαθέσιμες γωνίες και για 0 ανάμεσα στις γωνίες.
 - ii) Αν υπάρχει X σε ανάμεσα σε γωνίες τότε για 0 στις υπόλοιπες διαθέσιμες θέσεις ανάμεσα σε γωνίες και για 0 στις γωνίες
 - iii) Αν δεν υπάρχει X στο κενό τότε για 0 στο κενό και αν υπάρχει τότε για 0 στις γωνίες ή ανάμεσα στις γωνίες.



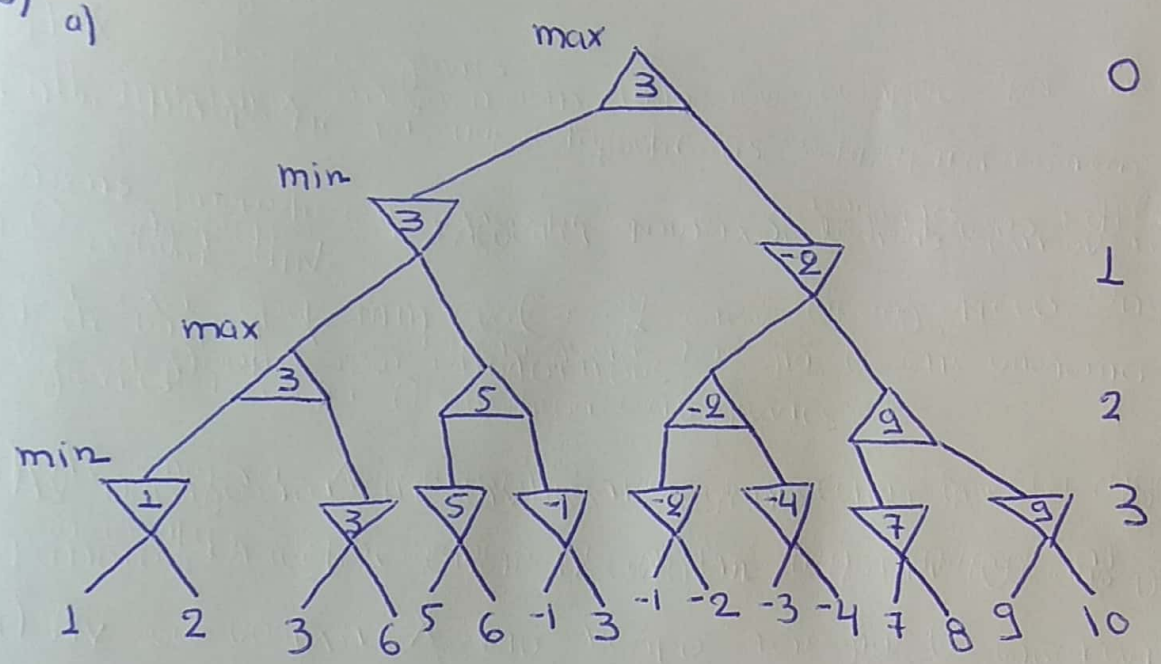
- 4. Η minimax απόφαση στη ρίζα του δέντρου θα είναι 1
- 5. Ο αλγόριθμος θα κτεδέψει τους 3 τελευταίους κόμβους επειδή θα έχει βρει min eval = -2 στον προηγούμενο. Οπότε δεν χρειάζεται να ελέγξουμε τους υπολοίπους.

Αν παράγονται με την αντίστροφη σειρά δεν θα πραγματοποιηθεί κάποιο κλάδεμα όπου η βέλτιστη είναι παραχμής είναι η αρχική.

2) Αν ξεκινήσουμε με τον max παίκτη η ιδιότητα που χρειάζεται ώστε το κλάδεμα των κόμβων να είναι μέγιστο είναι το πρώτο φύλλο κάθε min κόμβου να έχει χρησιμότητα μικρότερη από αυτήν του προηγούμενου κόμβου. Με αυτόν τον τρόπο θα κλαδεύεται κάθε φορά το υπόλοιπο του τωρινού κόμβου.

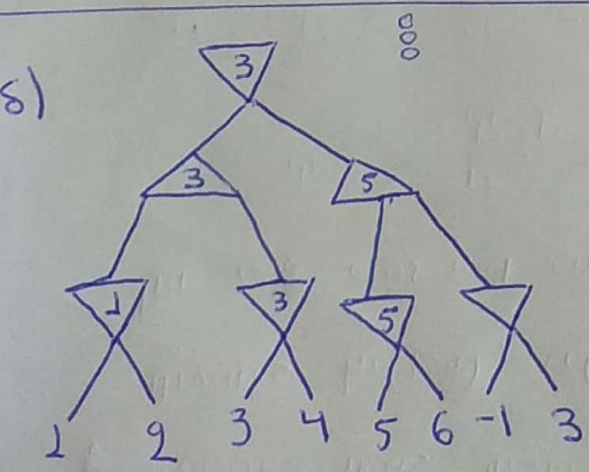
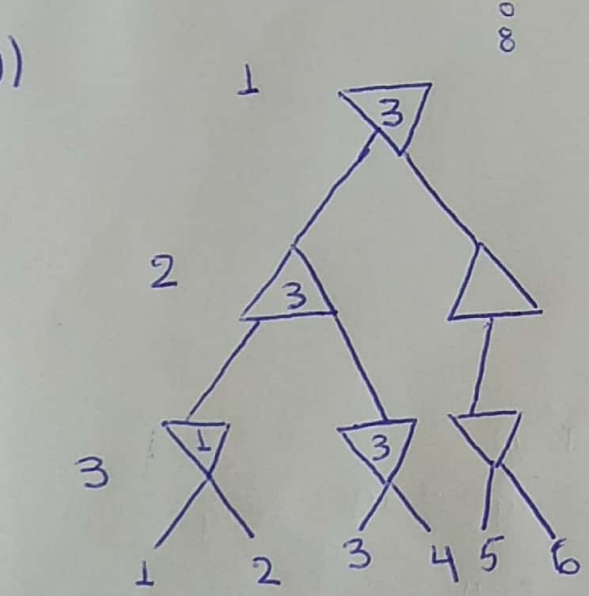
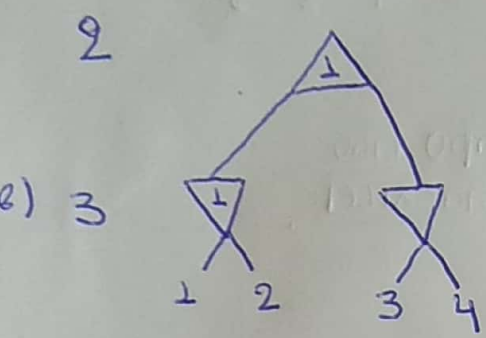
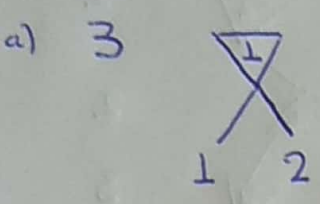
Αντίθετα για να είναι ελάχιστο θα πρέπει να βρίσκεται στα φύλλα του πρώτου κόμβου η χρησιμότητα με την μικρότερη τιμή. Θα δεν θα πραγματοποιηθεί κανένα κλάδεμα.

3) α)

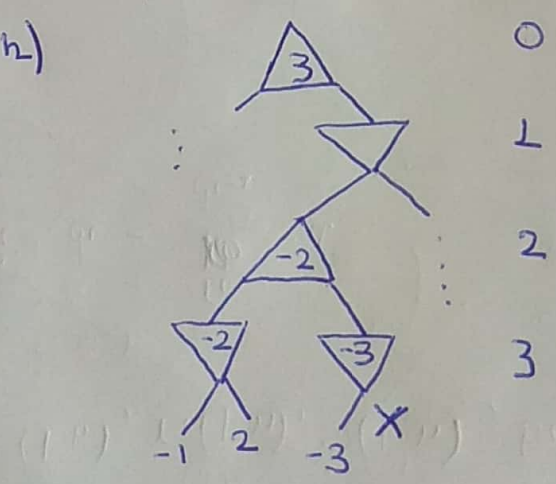
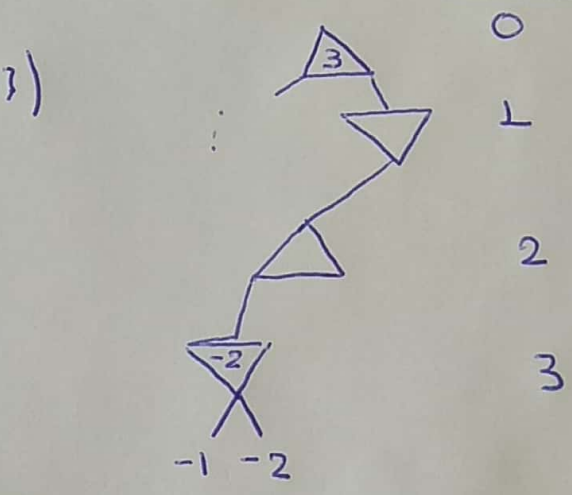
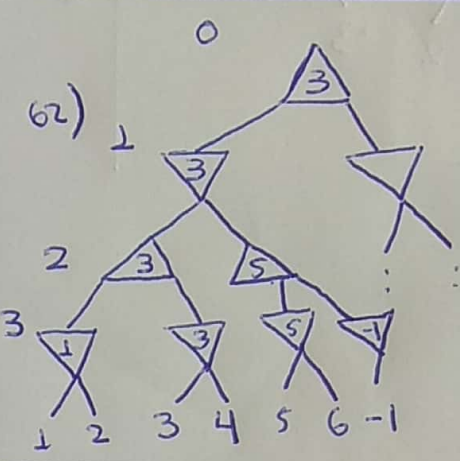
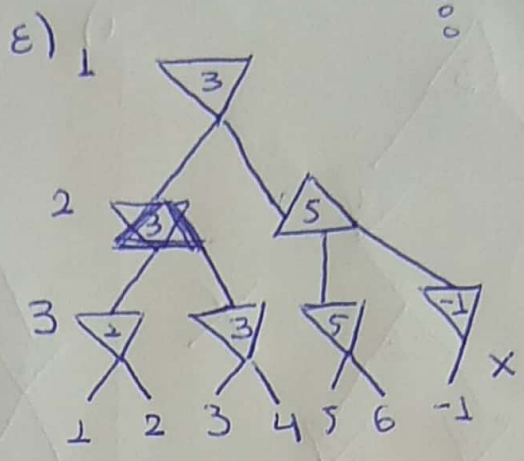


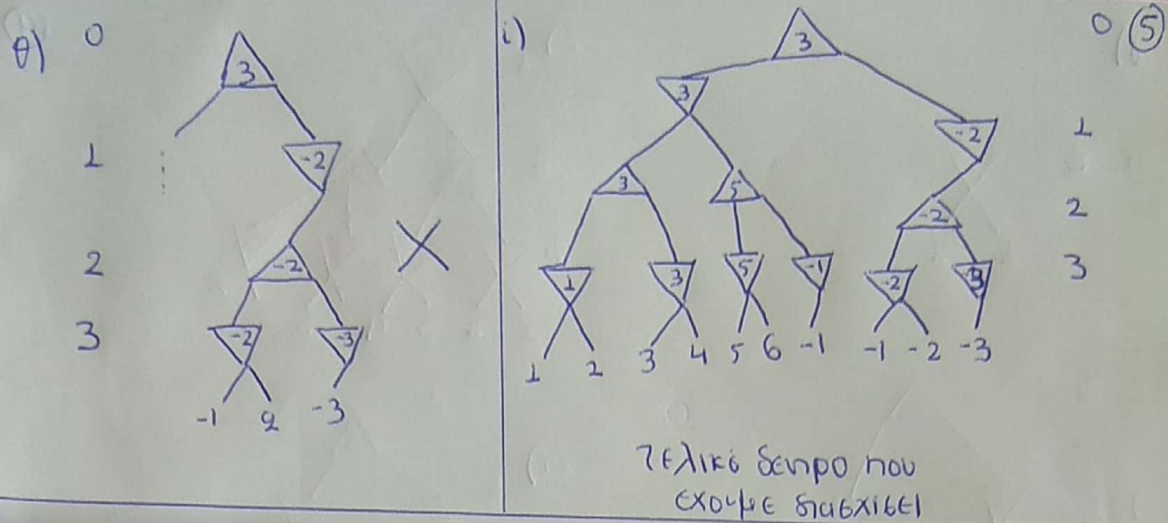
β) Η απόφαση είναι να ακολουθηθεί την διαδρομή στο δέντρο η οποία δίνει αποτέλεσμα 3.

7)



4





4. α) Αφού δεν υπάρχουν τμήματα κομβοί δεν έχω νόημα το κλάδεμα καθώς δεν θα επιλεγεί ποτέ χειρότερη τιμή από την προηγούμενη και ~~το~~ πάντα θα φαχνουμε όλους τους κομβούς για μια μεγαλύτερη τιμή.
- β) Όχι καθώς ο αλγόριθμος expectimax χρειάζεται να διακρίνει στο δέντρο για να υπολογίσει τους chance κομβούς καθώς μια τιμή χρησιμότητας μπορεί να αλλάξει το αποτέλεσμα.
- γ) Όταν βρούμε την τιμή 0 τότε δεν έχω νόημα να συνεχίσουμε να φαχνουμε το δέντρο γιατί έχουμε βρει την μεγαλύτερη, άρα κλαδεύουμε το υπόλοιπο.
- δ) Αν σε έναν κομβό chance έχουμε τιμή 0 ~~από~~ θα σημαίνει ότι όλες οι προηγούμενες τιμές ήταν 0 τότε μπορούμε να μην επισκεφτούμε το υπόλοιπο δέντρο καθώς θα έχουμε βρει την καλύτερη διαδρομή και μπορούμε να κλαδέψουμε το υπόλοιπο δέντρο.
- ε) Όχι δεν είναι δυνατό να πραγματοποιηθεί κλάδεμα και ο λόγος είναι ο ίδιος με το ερώτημα (α).
- 62) Παρόμοια θα γίνει και ο λόγος είναι ο ίδιος με το ερώτημα (β).
- 1) Αν σε ένα φύλλο βρούμε την τιμή 1 τότε παρόμοια με το ερώτημα (γ) έχουμε βρει την καλύτερη λύση οπότε κλαδεύουμε το υπόλοιπο.
- 2) Αν βρεθεί η τιμή 1 σε ένα chance κομβό τότε όπως και στο ερώτημα (δ) μπορούμε να κλαδέψουμε το υπόλοιπο δέντρο καθώς οι κομβοί που βρίσκονται κάτω από τον κομβό είναι όλοι 1, ~~και~~ οπότε θα έχουμε βρει και την καλύτερη λύση.