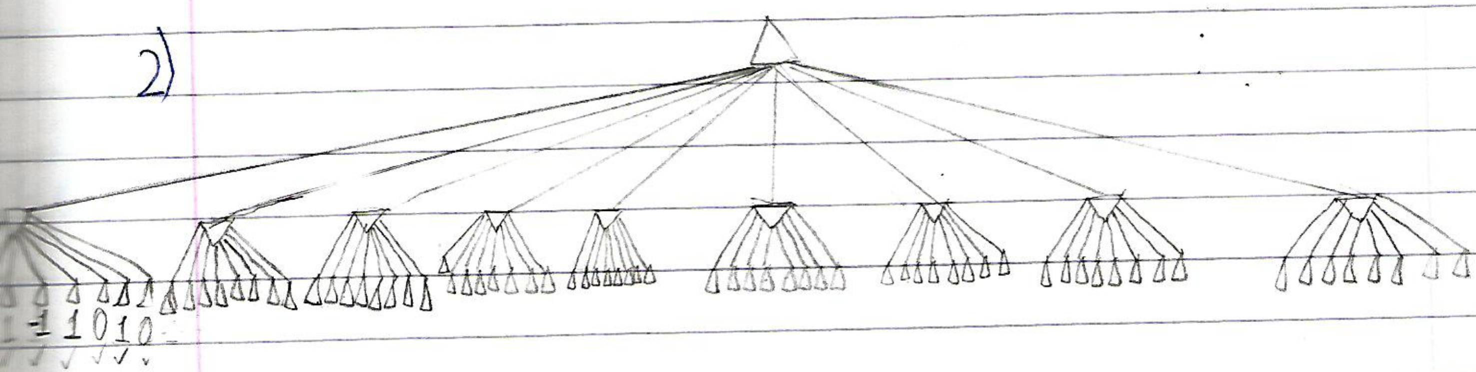
Κωνσταντίνος Σκορδούλης
1115 2016 00155

1	2	3
4	5	6
7	8	9

Project 2 AI

Πρόβλημα 2 Θεωρούμε πως $\boxed{Max = X}$, $\boxed{Min = O}$

2)



⊖ Εξήγηση: Υποθέτουμε ότι ξεκινάει ο X (Max-Player).
Αρχικά έχει 9 διαθέσιμες κινήσεις.
Μετά ο O (Min-Player) έχει 8 διαθέσιμες κινήσεις. *

* Π.χ. Αν ο Max επιλέξει το 1ο κελί και ο Min το 2ο, έχουμε το τέρμα αριστερά παιδί (βάθος 2)

3) Χρησιμοποιούμε την συνάρτηση αποτίμησης:

$Eval(s) = 3\cancel{X_2(s)}^{\,0} + X_1(s) - (3\cancel{O_2(s)}^{\,0} + O_1(s)) \Rightarrow$ (γιατί 1 κίνηση έχει κάθε Player)

$\Rightarrow \boxed{Eval(s) = X_1(s) - O_1(s)}$

* Τα γράφω εδώ πιο καθαρά (για το 1ο, 2ο, 3ο,.., Min)

α) 1, 0, 1, -1, 1, 0, 1, 0 ✓
β) -1, -1, 0, -2, 0, -1, 0, -1 ✓
γ) 0, 1, 1, -1, 1, 0, 1, 0 ✓
δ) -1, 0, -1, -2, 0, -1, 0, -1 ✓
ε) 1, 2, 1, 2, 2, 1, 2, 1 ✓
στ) -1, 0, -1, 0, -2, -1, 0, -1 ✓
ζ) 0, 1, 0, 1, -1, 1, 1, 0 ✓
η) -1, 0, -1, 0, -2, 0, -1, -1 ✓
θ) 0, 1, 0, 1, -1, 1, 0, 1 ✓

* Και οι δύο έχουν παίξει (φύλλα)

~~* (Επίπεδο Min) → Μόνο ο Max έχει παίξει.~~

~~α) 3 β) 2 γ) 3 δ) 2 ε) 4~~

~~στ) 2 ζ) 3 η) 2 θ) 3~~

~~* (Επίπεδο Max → Root) → Κανένας δεν έχει παίξει~~

~~$\boxed{Eval(s) = 0}$~~

7) Σύμφωνα με τις αποτιμήσεις των κόμβων (προηγούμενο ερώτημα) βρίσκω τις minimax τιμές των (Min) κόμβων.

Οπότε έχουμε: (από αριστερά στα δεξιά)

(α) (β) (γ) (δ) (ε) (στ) (ζ) (η) (θ)

-1 , -2, -1, -2 , 1, -2, -1 , -2 , -1

Θ Από αυτές τις τιμές, επιλέγω την $\boxed{\text{maximum} \Rightarrow 1}$

↳ Άρα το (Max) node παίρνει την τιμή 1

Θ Άρα ο Max-Player (χ) θα επιλέξει το action που θα μας οδηγήσει στον Min-κόμβο (ε) που έχει τιμή 1.

Θ Αυτό γίνεται γιατί ο Minimax αλγόριθμος υποθέτει πως οι Max-Min agents θα κάνουν τις optimal κινήσεις. Άρα σε αυτές τις worst case συνθήκες, βρίσκουμε την κίνηση/action που μεγιστοποιεί το κέρδος του (Max) agent.

⊕ Επειδή είναι 2 ply game (1 Max, 1 Min) => δεν ασχολούμαστε με το b στον περιορισμό

5) a) Left -> Right:

⊖ Αρχίζουμε το Max-node το οποίο έχει $[-\infty, +\infty] \Rightarrow a = -\infty,\ b = +\infty$
Κατεβαίνουμε στον 1ο Min κόμβο

1o Min ⊖ Επειδή το $a = -\infty$, δεν θα υπάρξει εδώ κλάδεμα.
Απλά διαμορφώνεται η τιμή του v: (Ψάχνουμε σε terminal states)

$1, 0, 0, -1, -1, -1, -1, -1 \Rightarrow$
⊕ Ο 1ος Min κόμβος έχει την τιμή $-1 \Rightarrow [-1, -1]$
⊕ Διαμορφώνουμε την τιμή του Max κόμβου $\Rightarrow [-1, +\infty] \Rightarrow \boxed{a = -1}, \boxed{b = +\infty}$

2o Min ⊖ Στο 2o Min-node, κατεβαίνει η τιμή $a = -1$ και κοιτάμε 1-1 τους κόμβους
Οπότε για το v του τωρινού Min-Node έχουμε:

$-1, -1, -1, -2 \Rightarrow$ STOP!! $\Rightarrow \boxed{-2 < a = -1}$
⊕ Pruned-Nodes: 5, 6, 7, 8 (δηλαδή οι μισοί) $\Rightarrow \# = 4$
⊕ Το 2o Min-Node θα έχει τιμή $-2 \Rightarrow [-\infty, -2]$ => δεν θα το επιλέξει ο Max

3o Min ⊖ Στο 3o Min-node, πάλι κατεβαίνει $a = -1$
Πάλι κοιτάζουμε τους κόμβους και οι τιμές του v είναι:

$0, 0, 0, -1, -1, -1, -1, -1$ => δεν κλαδέψαμε τίποτα
⊕ Ο 2ος Min κόμβο έχει τιμή $-1 \Rightarrow [-1, -1]$

4o Min ⊖ Στο 4o Min, πάλι κατεβαίνει $a = -1$
Η τιμές του v θα είναι:

$-1, -1, -1, -2 \Rightarrow$ STOP!! $\Rightarrow \boxed{-2 < a = -1}$
⊕ Pruned-Nodes: 5, 6, 7, 8 (δηλαδή οι μισοί) $\Rightarrow \# = 4$
⊕ Το 4o Min-Node θα έχει τιμή $-2 \Rightarrow [-\infty, -2]$ => δεν θα το επιλέξει ο Max

(+) Total Pruned Nodes: 4+4+7+7+7+7= 8+28=(36) από τα 72 (!)

(5o Min)

(−) Στο 5ο Min-Node, πάλι κατεβαίνει (a=−1), και οι τιμές του v είναι:

1, 1, 1, 1, 1, 1, 1, 1 => Δεν κλαδέψαμε τίποτα

(+) Η τιμή του 5ου Min-Node είναι (1) => [1, 1]

(+) Τώρα στο (Max) έχουμε [1, +∞] => $\boxed{a=1}$, $\boxed{b=+\infty}$

(6o Min) (−) Στο 6ο Min Node, κατεβαίνει (a=1) και οι τιμές του v είναι:

(−1) => STOP!! => $\boxed{-1 < a=1}$

(+) Η τιμή του 6ου Min-Node είναι (−1) => [−∞, −1]

(+) Pruned Nodes: 2, 3, 4, 5, 6, 7, 8 (όλοι εκτός από το 1ο) => # = 7

(7o Min) (−) Στο 7ο Min Node, κατεβαίνει (a=1) και οι τιμές του v είναι:

(0) => STOP!! => $\boxed{0 < a=1}$

(+) Pruned Nodes: 2, 3, 4, 5, 6, 7, 8 (όλοι εκτός από το 1ο) => # = 7

(+) Η τιμή του 7ου Min Node είναι (0) => [−∞, 0]

(8o Min) (−) Στο 8ο Min Node, κατεβαίνει (a=1) και οι τιμές του v είναι:

(−1) => STOP!! => $\boxed{-1 < a=1}$

(+) Pruned Nodes: 2, 3, 4, 5, 6, 7, 8 (όλοι εκτός από το 1ο) => # = 7

(+) Η τιμή του 8ου Min Node είναι (−1) => [−∞, −1]

(9o Min) (−) Στο 9ο Min Node, κατεβαίνει (a=1) και οι τιμές του v είναι:

(0) => STOP!! => $\boxed{0 < a < 1}$

(+) Pruned Nodes: 2, 3, 4, 5, 6, 7, 8 (όλοι εκτός από το 1ο) => # = 7

(+) Η τιμή του 9ου Min Node είναι (0) => [−∞, 0]

(Finale) (−) Από τις τιμές: −1, −2, −1, −2, (1), −1, 0, −1, 0 => Επιλέγει τον 5ο Min κόμβο

=> (X) στη θέση (2x2) ← κέντρο

⇒ Αλλαγή όταν λέω 9,8,7,6 ⇒ 1,2,3,4
(Right->Left) (Left->Right)

④ Αλλάζω όλους κόμβους

β) Right -> Left: Αντίστροφη σειρά

(*) Αυτή τη φορά θα τα γράψω πιο συνοπτικά, αρχίζοντας από το 9ο Min-Node και κοιτάζοντας τους τερματικούς κόμβους από δεξιά->αριστερά

9ο Min ⊖ $a=-\infty$, και οι τιμές του v:

$1,0,0,-1,-1,-1,-1,-1 \Rightarrow \boxed{Minimax=-1} \Rightarrow \boxed{a=-1}$
$\Downarrow [-1,-1]$

8ο Min ⊖ $a=-1$, και οι τιμές του v:

$-1,-1,-1,-2 \Rightarrow$ STOP!! $\Rightarrow \boxed{-2<a=-1}, \boxed{Minimax=-2} \Rightarrow [-\infty,-2]$
⊕ Pruned Nodes: 5,6,7,8 (τα μισά) (*)

7ο Min ⊖ $a=-1$, και οι τιμές του v:

$0,0,0,-1,-1,-1,-1,-1 \Rightarrow \boxed{Minimax=-1} \Rightarrow [-1,-1]$

6ο Min ⊖ $a=-1$ και οι τιμές του v:

$\boxed{-2<a=-1}$
$-1,-1,-1,-2 \Rightarrow$ STOP!! $\Rightarrow \boxed{Minimax=-2} \Rightarrow$ Pruned Nodes: 5,6,7,8

5ο Min ⊖ $a=-1$ και οι τιμές του v:

$1,1,1,1,1,1,1,1 \Rightarrow \boxed{Minimax=1} \Rightarrow \boxed{a=1}$
$\Downarrow [1,1]$

4ο Min ⊖ $a=1$ και οι τιμές του v:

$\boxed{-1<a=1}$
$-1 \Rightarrow$ STOP!! $\Rightarrow \boxed{Minimax=-1} \Rightarrow$ Pruned Nodes: 2,3,4,5,6,7,8

3ο Min ⊖ $a=1$ και οι τιμές του v:

$0 \Rightarrow$ STOP!! $\Rightarrow \boxed{0<a=1} \Rightarrow \boxed{Minimax=0} \Rightarrow$ Pruned Nodes: 2,3,4,5,6,7,8

Total Pruned Nodes: 4+4+7+7+7+7 = 36 από 72

i) α=1 και οι τιμές του v είναι:

-1 => STOP!! => $\boxed{-1<\alpha=1}$ => $\boxed{Minimax=-1}$ => Pruned Nodes: 2,3,4,5,6,7,8

j) α=1 και οι τιμές του v είναι:

0 => STOP!! => $\boxed{0<\alpha=1}$ => $\boxed{Minimax=0}$ => Pruned Nodes: 2,3,4,5,6,7,8

k) Από τις τιμές: -1, -2, -1, -2, 1, -1, 0, -1, 0

+ Επιλέγω τον 5ο Min κόμβο => Ίδιο αποτέλεσμα με πριν

+ Επιπλέον $\boxed{\text{ίδιο \# Pruned Nodes}}$

γ) Βέλτιστη Σειρά παραγωγής:

Θ Βέλτιστη σειρά, θεωρούμε αυτή που μεγιστοποιεί το πλήθος των κλαδεμένων (pruned) κόμβων.

Θ Έστω ότι τα φύλλα ενός κόμβου Min, αποτελούν 1 ακολουθία S
Π.χ 1, 3, -1, 7, 12, 4, 0, 6

⊕ Έστω a=0, θα πρέπει να φτάσουμε στο -1 πριν διακόψουμε την αναζήτηση (κλαδεύω 5 κόμβους)

Θ Μια ιδέα είναι: Να ταξινομήσω την ακολουθία ως μη-φθίνουσα ακολουθία

⊕ Δηλαδή το πρώτο φύλλο που θα επισκέπτομαι να είναι το minimum της ακολουθίας.

⊕ Με αυτό το τρόπο, είτε θα επισκέπτομαι το 1ο μόνο είτε όλα τα φύλλα (οπότε θα μπορούμε γρήγορα να αποφασίζουμε αν είναι να κλαδέψουμε το κόμβο

Πρόβλημα 2

⊖ Maximum Pruning: Οι τιμές χρησιμότητας πρέπει να είναι μία μη φθίνουσα ακολουθία. Δηλαδή να εξερευνούμε τον minimum των φύλλων, πρώτο.

⊕ Έτσι κοιτάμε όσο το δυνατόν λιγότερους κόμβους. Είτε θα εξετάσει μόνο το 1ο είτε όλου

⊖ Minimum Pruning: Το αντίθετο, τώρα θέλουμε μία μη αύξουσα ακολουθία. Δηλαδή ο πρώτος κόμβος να είναι ο maximum ενώ ο τελευταίος ο minimum

⊕ Έτσι θα κοιτάμε όσο το δυνατόν περισσότερα φ

Πρόβλημα 3

α) Οι Minimax τιμές, από κάτω προς τα πάνω, είναι: (level by level)

2ο Min ⊖	1	3	5	-1	-2	-4	7	10
2ο Max ⊖	3		5		-2		10	
1ο Min ⊖	3				-2			
Max ⊖	3							

β) Η Minimax απόφαση του δέντρου είναι να επιλέξει το 1ο action, δηλαδή τον αριστερό κόμβο γ) Α) Αρχίζουμε με το Root-Max, όπου $\boxed{a=-\infty}$, $\boxed{b=+\infty}$, $\boxed{v=-\infty}$

Min B1) Κατεβαίνουμε στο αριστερό παιδί-Min, με $\boxed{a=-\infty}$, $\boxed{b=+\infty}$, $\boxed{v=+\infty}$

Max Γ1) Το ίδιο για το αριστερό Max-παιδί, με $\boxed{a=-\infty}$, $\boxed{b+\infty}$, $\boxed{v=-\infty}$

Min Δ1) Τώρα φτάσαμε στον προτελευταίο-αριστερό Min κόμβο, $\boxed{a=-\infty}$ $\boxed{b=+\infty}$ $\boxed{v=-\infty}$

⊖ Οι τιμές του v, και του b, θα είναι: (v, b)

(1, 1), (1, 1) ✓ => $\boxed{b=1}$, $\boxed{a=-\infty}$ => Return $\boxed{v=1}$

Max Γ1) Γυρνάμε στον Γ1 => $\boxed{a=1}$, $\boxed{b=+\infty}$, $\boxed{v=1}$

Min Δ2) Επισκεπτόμαστε το άλλο παιδί με $\boxed{a=1}$, $\boxed{b=+\infty}$, v=+∞

⊖ Οι τιμές του v, και του b, θα είναι: (v, b)

(3, 3), (3, 3) ✓ => $\boxed{b=3}$, $\boxed{a=1}$ => Return $\boxed{v=3}$

Max Γ1) Επιστρέφουμε πάλι στο Γ1 => $\boxed{a=1}$, $\boxed{b=+\infty}$ => Return $\boxed{v=3}$

Min B1) Επιστρέφουμε στο B1 με $\boxed{a=-\infty}$, $\boxed{b=3}$, $\boxed{v=3}$

Max Γ2) Τώρα στο δεξί παιδί με $\boxed{a=-\infty}$, $\boxed{b=3}$, $\boxed{v=-\infty}$

Min Δ3) Στο αριστερό παιδί, με $\boxed{a=-\infty}$, $\boxed{b=3}$, $\boxed{v=+\infty}$

⊖ Οι τιμές του v, και του b, θα είναι: (v, b)

5 => **STOP!!** => $\boxed{v=5>b=3}$ => Return $\boxed{v=5}$

⊕ Pruned Nodes: E6 ✓

(Max) Γ2) Επιστρέφουμε στο (Γ2):

⊖ Το $\boxed{v = 5 > b = 3}$ ⇒ STOP!! ⇒ $\boxed{a = -\infty}, \boxed{b = 3}$ ⇒ Return $\boxed{v = 5}$

⊕ Pruned Nodes: (Δ4) → (E7), (E8)

(Min) B1) Επιστρέφουμε στο (B1) με $\boxed{a = -\infty}, \boxed{b = 3}$ $\boxed{v=3}$ ⇒ Return $\boxed{v = 3}$

(Max) A) Επιστρέφουμε στο (Root) με $\boxed{a = 3}, \boxed{b = +\infty}, \boxed{v = 3}$

(Min) B2) Κατεβαίνουμε στο δεξί παιδί με $\boxed{a = 3}, \boxed{b = +\infty}, \boxed{v = +\infty}$

(Max) Γ3) Way down we go, $\boxed{a = 3}, \boxed{b = +\infty}, \boxed{v = -\infty}$

(Min) Δ5) Φτάσαμε στο Δ5, $\boxed{a = 3}, \boxed{b = +\infty}, \boxed{v = +\infty}$

⊖ Οι τιμές του v, και του b, είναι:

$\boxed{v = -1 < a = 3}$ ⇒ STOP!! ⇒ $\boxed{a = 3, b = +\infty}$ ⇒ Return $\boxed{v = -1}$

⊖ Pruned Nodes: (E10)

(Max) Γ3) Επιστρέφουμε στο Γ3 με $\boxed{a = 3}, \boxed{b = +\infty}, \boxed{v = -1}$

(Min) Δ6) Κατεβαίνουμε στο δεξί παιδί, $\boxed{a = 3}, \boxed{b = +\infty}$:

⊖ Οι τιμές του v, και του b, είναι:

$\boxed{v = -3 < a = 3}$ ⇒ STOP!! ⇒ $\boxed{a = 3}, \boxed{b = +\infty}$ ⇒ Return $\boxed{v = -1}$

⊖ Pruned Nodes: (E12)

Max Γ3) Επιστρέφουμε στο Γ3 ⇒ $\boxed{a=3}$, $\boxed{b=+\infty}$, $\boxed{v=-1}$ ⇒ Return $v=-1$

B2) Επιστρέφουμε στο B2 που είχε $\boxed{a=3}$, $\boxed{b=+\infty}$:

$\boxed{v=-1 < a=3}$ ⇒ STOP!! ⇒ $\boxed{a=3}$, $\boxed{b=+\infty}$ ⇒ Return $v=-1$

A) Επιστρέφοντας στο Root: $\boxed{a=3}$, $\boxed{b=+\infty}$, $\boxed{v=3}$

⊖ Επομένως ο Max, θα επιλέξει την ενέργεια/action με τιμή $v=3$ ⇒ δηλαδή το $\boxed{\text{αριστερό παιδί}}$!

⊖ Pruned Nodes: a) Non-leaves: Δ4, Γ4 → Δ7, Δ8

b) Leaves: E6, E7, E8, E10, E12, E13, E14, E15, E16

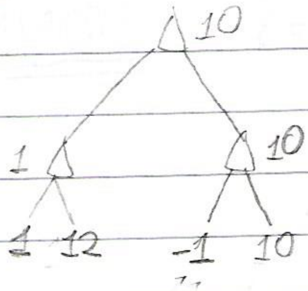
⊕ Συνολικά: $\boxed{\text{9 από 16 κόμβοι κλαδεύτηκαν}}$

Πρόβλημα 4

α) Μια ιδέα είναι να έχουμε τη παράμετρο max, η οποία θα κρατάει τον max-αριθμό που έχουμε βρει μέχρι τώρα.
Όταν βρίσκει κάποιο μεγαλύτερο από max, να επιστρέφει (pruning)

⊖ Όμως υπάρχει η περίπτωση:

⊕ Αρχίζουμε $\boxed{max = -\infty}$



⊕ Με αυτό το σκεπτικό, επειδή $\boxed{v = 1 > max = -\infty}$, θα κλαδέψει το 2ο φύλλο, και έτσι οδηγούμαστε σε λάθος αποτέλεσμα.

⇒ Λάθος αποτέλεσμα ⇒ ΑΔΥΝΑΤΟΝ

β) ΑΔΥΝΑΤΟΝ

⊖ Τώρα που έχουμε Chance Nodes, μας ενδιαφέρει και η πιθανότητα να εκτελέσουμε μια ενέργεια

$$\boxed{\sum_{i=1}^{n} P(i)\, Expectimax(i)} \quad , \quad \boxed{0 < P(i) \le 1}$$

⊖ Επειδή έχουμε άθροισμα και κανένα φράγμα ως προς τις τιμές των φύλλων (ούτε γνωρίζουμε κάτι για τα P(i) ⇒ Κατανομή(?))

⇒ Τεράστια διακύμανση τιμών (Η προσθήκη ή κλάδεμα μπορεί να αλλάξει το αποτέλ.)

⇒ Δεν μπορούμε να κάνουμε πρώιμες προβλέψεις, μέχρι να εξετάσουμε όλα τα φύλλα γ) Οι τιμές των φύλλων βρίσκονται στο διάστημα $(-\infty, 0]$

⊖ Επειδή μιλάμε για πραγματικούς αριθμούς, ο μόνος τρόπος κλαδέματος είναι ο εξής:

=> Αν κατά την αναζήτηση, συναντήσουμε κάποιο φύλλο με τιμή 0
=> Σταματάμε την αναζήτηση (και κλαδεύουμε τους υπόλοιπους κόμβους)

=> ΕΦΙΚΤΟ, μόνο αν βρούμε την τιμή 0 σε κάποιο φύλλο

Άνω φράγμα

δ) Έχουμε παρόμοια προβλήματα όπως στο (β)

⊖ Το θετικό είναι ότι φράσσονται οι τιμές των φύλλων στο $[-\infty, 0]$

⊖ Τα αρνητικά που εξακολουθούμε να έχουμε:
i) Ακόμα δεν γνωρίζουμε τίποτα για την κατανομή των P(i)

ii) Πάλι μιλάμε για άθροισμα

$\boxed{\sum_{i=1}^{n} P(i)\ \text{Expectimax}(i)}$, $\boxed{P(i) \cdot \text{Expectimax}(i) \leq 0}$

⊕ Κάθε φορά που προσθέτω έναν όρο (του αθροίσματος)
=> Μειώνω την τιμή (v) του Chance Node, με αποτέλεσμα να είναι λιγότερο πιθανό να επιλεγεί απο Max Node
(ή παραμένει ίδια)

⊕ Άρα προσθέτοντας ή κλαδεύοντας 1 κόμβο, πάλι μπορεί να αλλάξει αρκετά το αποτέλεσμα

=> ΑΔΥΝΑΤΟ

ε) $[0, +\infty)$. Τώρα δεν έχουμε άνω φράγμα, όπως στο γ, και επειδή έχουμε να κάνουμε με πραγματικούς αριθμούς (≥ 0)

=> Αντιμετωπίζουμε τα ίδια προβλήματα με το α

=> Δεν μπορούμε να κάνουμε πρώιμες προβλέψεις, γιατί κλαδεύοντας 1 κόμβο μπορεί να οδηγηθούμε σε λάθος αποτέλεσμα

=> ΑΔΥΝΑΤΟ

στ) ΑΔΥΝΑΤΟ => Ίδια προβλήματα με το δ

⊖ Η μόνη διαφορά είναι: $\boxed{P(i)\ \mathrm{Expectimax}(i) \geq 0}$

⊕ Προσθέτοντας ή κλαδεύοντας όρους του αθροίσματος, μπορεί να αλλάξει αρκετά το αποτέλεσμα
=> Μη επιθυμητό

ζ) Εφικτό => Όπως και στο γ, υπάρχει άνω φράγμα
↳ το 1

⊖ Αν κατά την αναζήτησή μας, βρούμε κόμβο με τιμή 1 τότε σταματάμε την αναζήτησή μας (δεν υπάρχει κόμβος με μεγαλύτερη τιμή)

Εφικτό

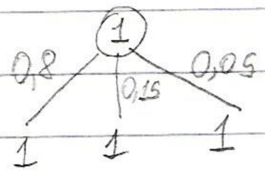
η) Καταρχάς έχουμε: $\boxed{0 \le P(i) \cdot Expectimax(i) \le 1}$

$$\boxed{0 \le \sum_{i=1}^{n} P(i) \cdot Expectimax(i) \le n}$$

⊖ Κάνοντας την υπόθεση ότι όλα τα Expectimax=1
και γνωρίζοντας ότι:

$$\boxed{\sum_{i=1}^{n} P(i) = 1} \Rightarrow \boxed{0 \le \sum_{i=1}^{n} P(i) \cdot Expectimax(i) \le 1}$$

⊛ Αυτό φαίνεται καλύτερα σε παράδειγμα:



$\Rightarrow$ $\boxed{Expectimax = 0,8 + 0,15 + 0,05 = 1}$

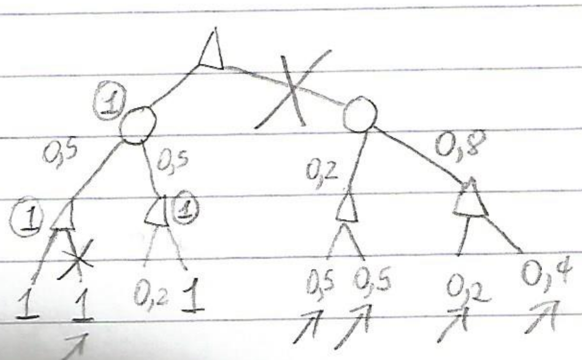
⊖ Οπότε η ιδέα είναι:

1) Chance

⊕ Αν κάποιο από τα παιδιά ενός Max Node ⇒ Chance ή Φύλλα

⇒ έχουν την τιμή 1 (max τιμή), δεν χρειάζεται να ψάξουμε τα υπόλοιπα παιδιά ⇒ Pruning

⊕ Παράδειγμα:



⊛ Επισκεφτήκαμε μόνο τα φύλλα 1, 3, 4