

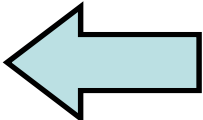


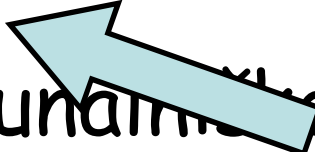
Kako računalnik naučimo igrati "4 v vrsto"

Branko Kavšek

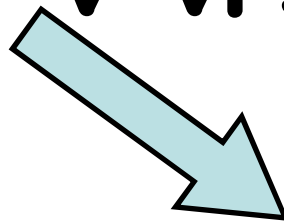
Izleti v matematično vesolje

Koper, Oktober 2012

Kaj bomo danes slišali?

- Kaj je "4 v vrsto"?  Kako igramo? = pravila igre
- Strategije igranja:  Kako zmagamo?
 - z upoštevanjem trenutne in naslednje poteze
 - z upoštevanjem vseh potez
- Kaj pa računalnik?  Strategije zapišemo kot program
- Kje je tu umetna inteligenca?
- Primer računalniške igrice "4 v vrsto"  Program se sam uči novih strategij

Kaj je "4 v vrsto"?



Angleško ime = **Connect 4**

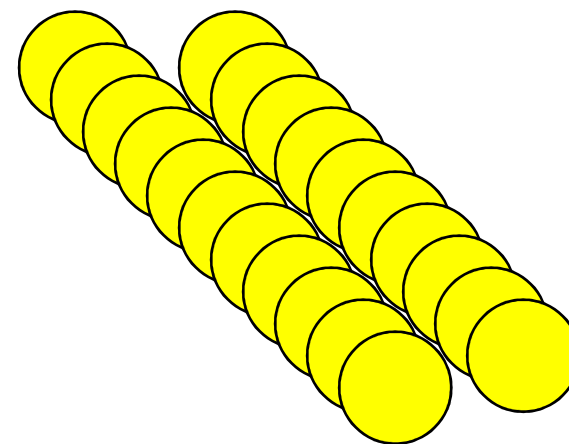
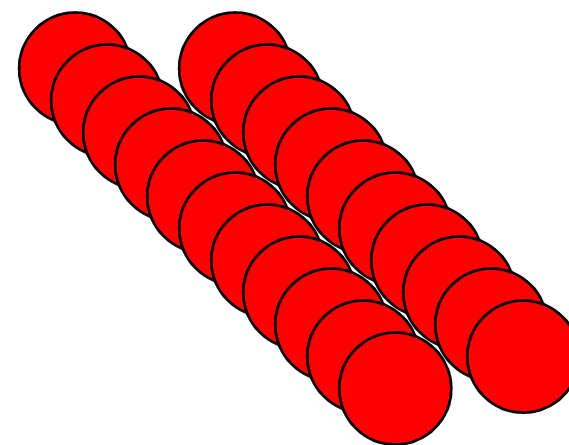
- Igra za 2 igralca
- Igralca se pri igranju izmenjujeta
(1. igralec na potezi, 2. igralec na potezi, 1. igralec, 2. igralec ...)
 - potezna igra
- Za igranje potrebujemo:
 - "mrežo" velikosti 7 x 6
 - 42 žetonov dveh barv (po 21 vsake barve)

Iz zgodovine

- Igro je igral že kapitan James Cook s svojimi oficirji na potovanjih po svetu
- Od tu je tudi dobila nadimek "Captain's Mistress" - "Kapitanova ljubica"
- Današnjo različico je uvedel Milton Bradley leta 1974 (Connect 4)

Igralne potreščine

6							
5							
4							
3							
2							
1							
	a	b	c	d	e	f	g



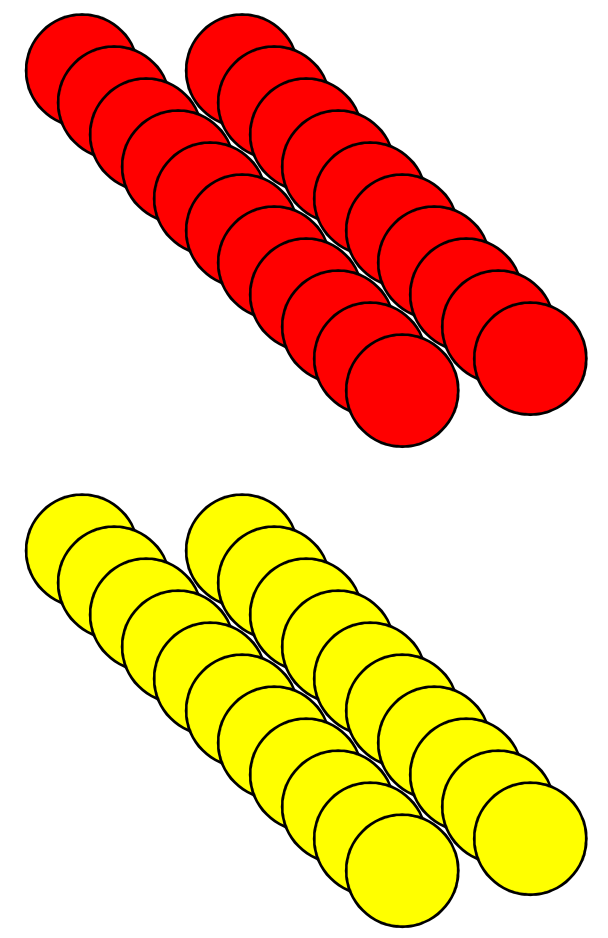
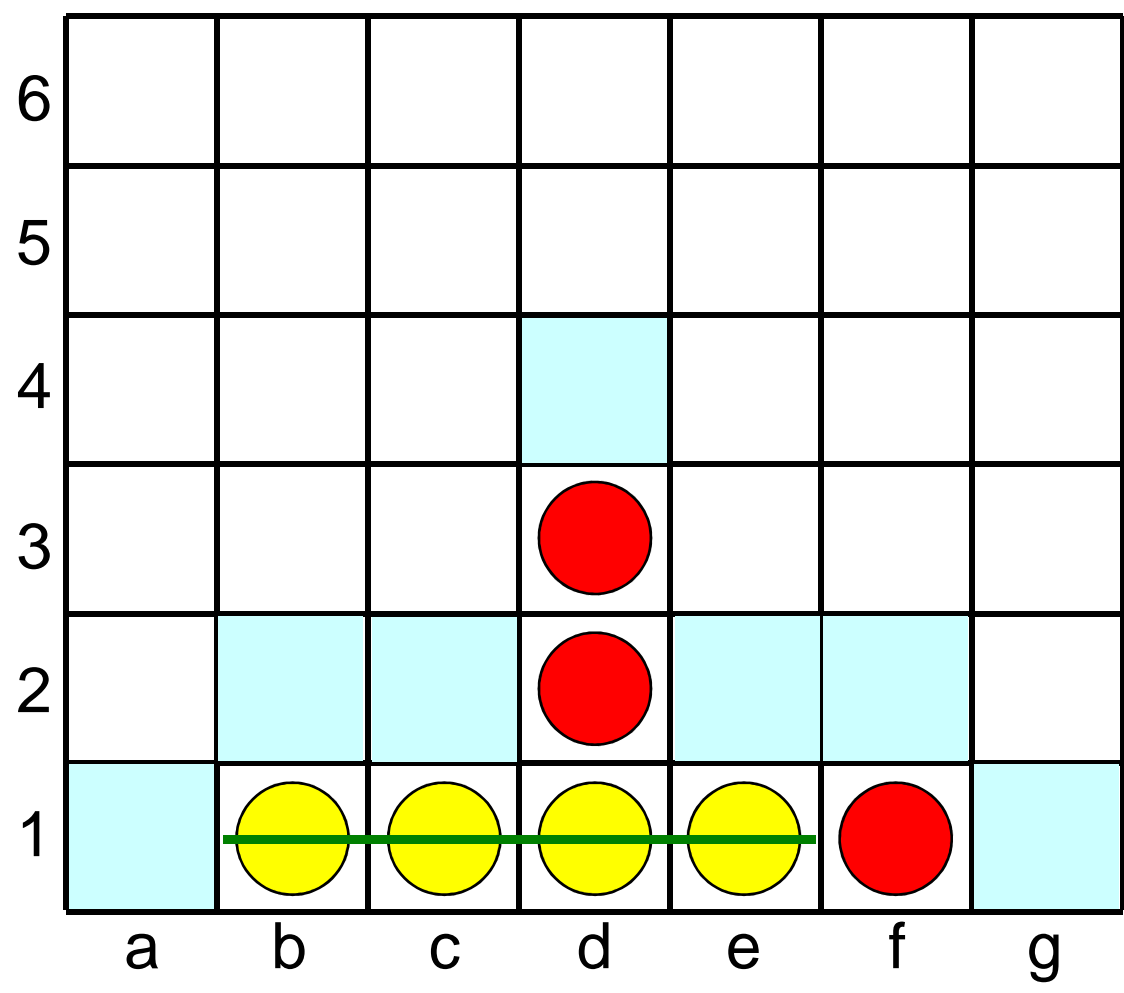
Pravila igre

- Začetek igre:
 - Igralca izbereta vsak svojo barvo žetonov
 - V prazno mrežo prvi igralec vrže žeton
 - različica: začne igralec z rumenimi žetoni
- Potek igre:
 - Igralca izmenično "mečeta" žetone v mrežo (poteza)
 - Žeton vedno "pade":
 - na dno mreže
 - na drugi žeton, ki je že v mreži

Konec (cilj) igre

- Igra se konča, ko:
 1. Eden od igralcev "poveže" 4 žetone svoje barve v vrsto (vodoravno, navpično ali diagonalno)
 2. Oba igralca porabita vse svoje žetone
- Možni izidi:
 - Zmaga (točka 1.)
 - Poraz (točka 1.)
 - Neodločeno (točka 2.)
- Cilj igre?
 - povezati 4 žetone svoje barve v vrsto

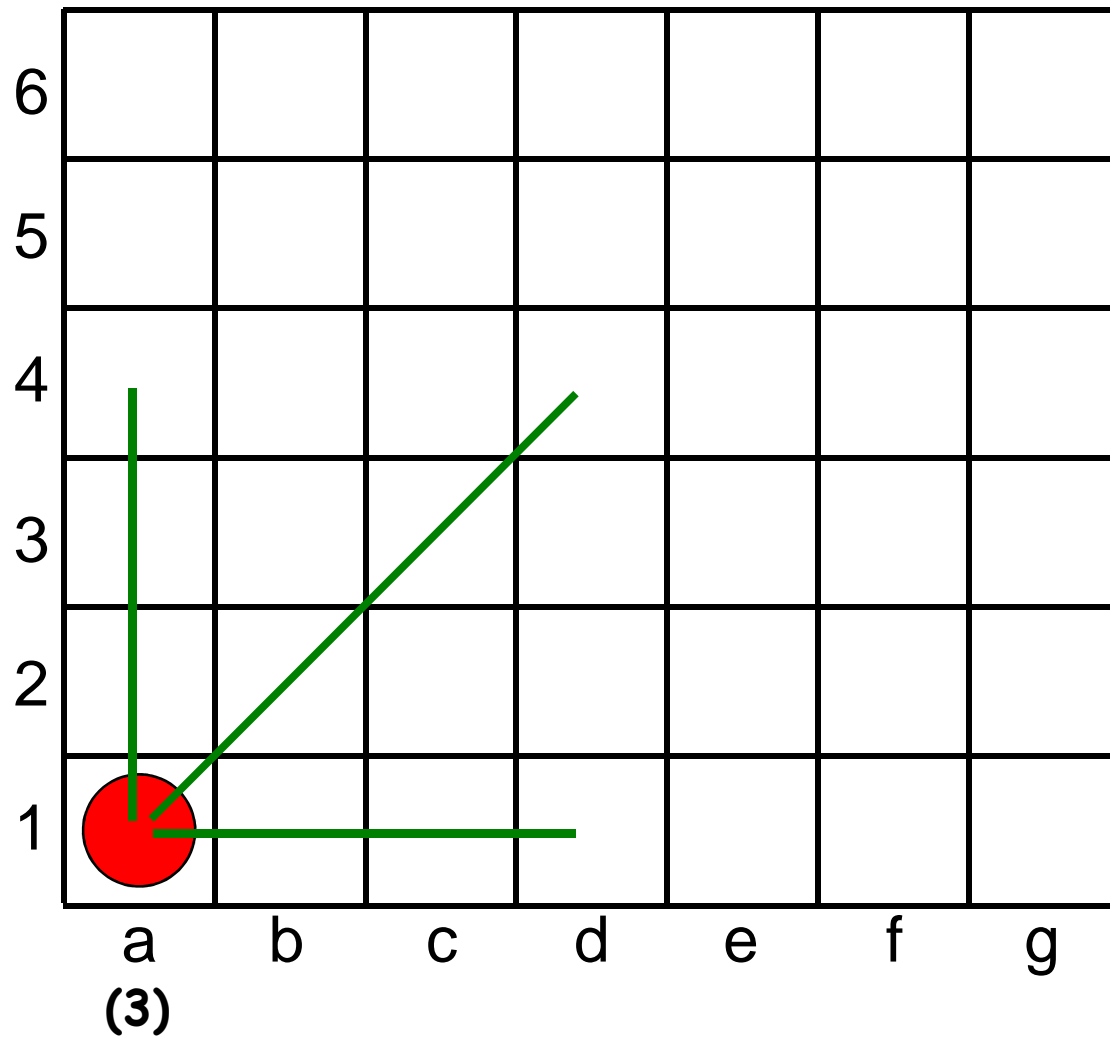
Primer igre



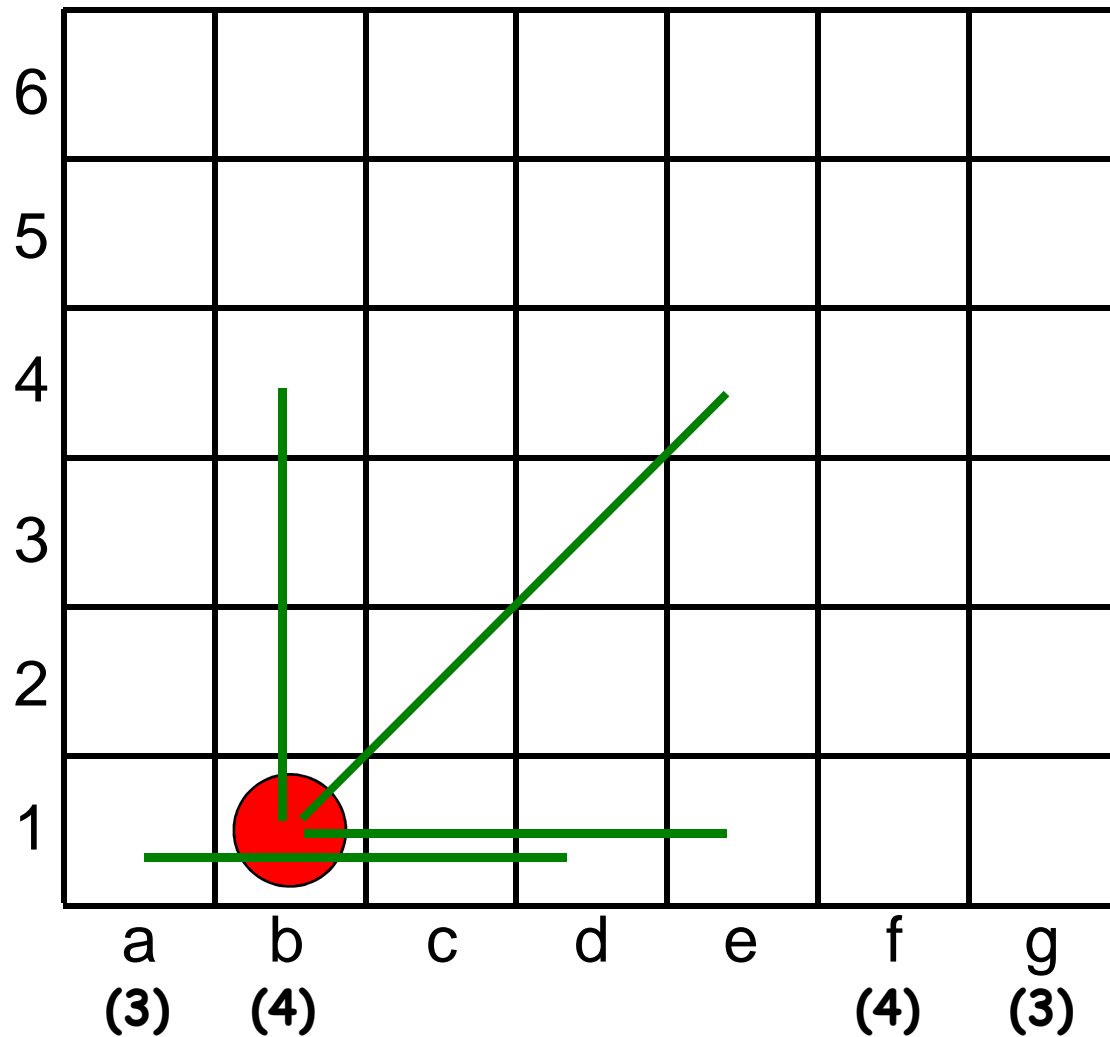
Kako zmagati igro? (1)

- Z upoštevanjem trenutne in naslednje poteze (strategije):
 1. Število možnih četvork
 2. Trenutno število žetonov v vrsti
 3. Ali zaprem nasprotnikovo "dobro" potezo?
 4. Ali prostor "zgoraj" prinese nasprotniku prednost?

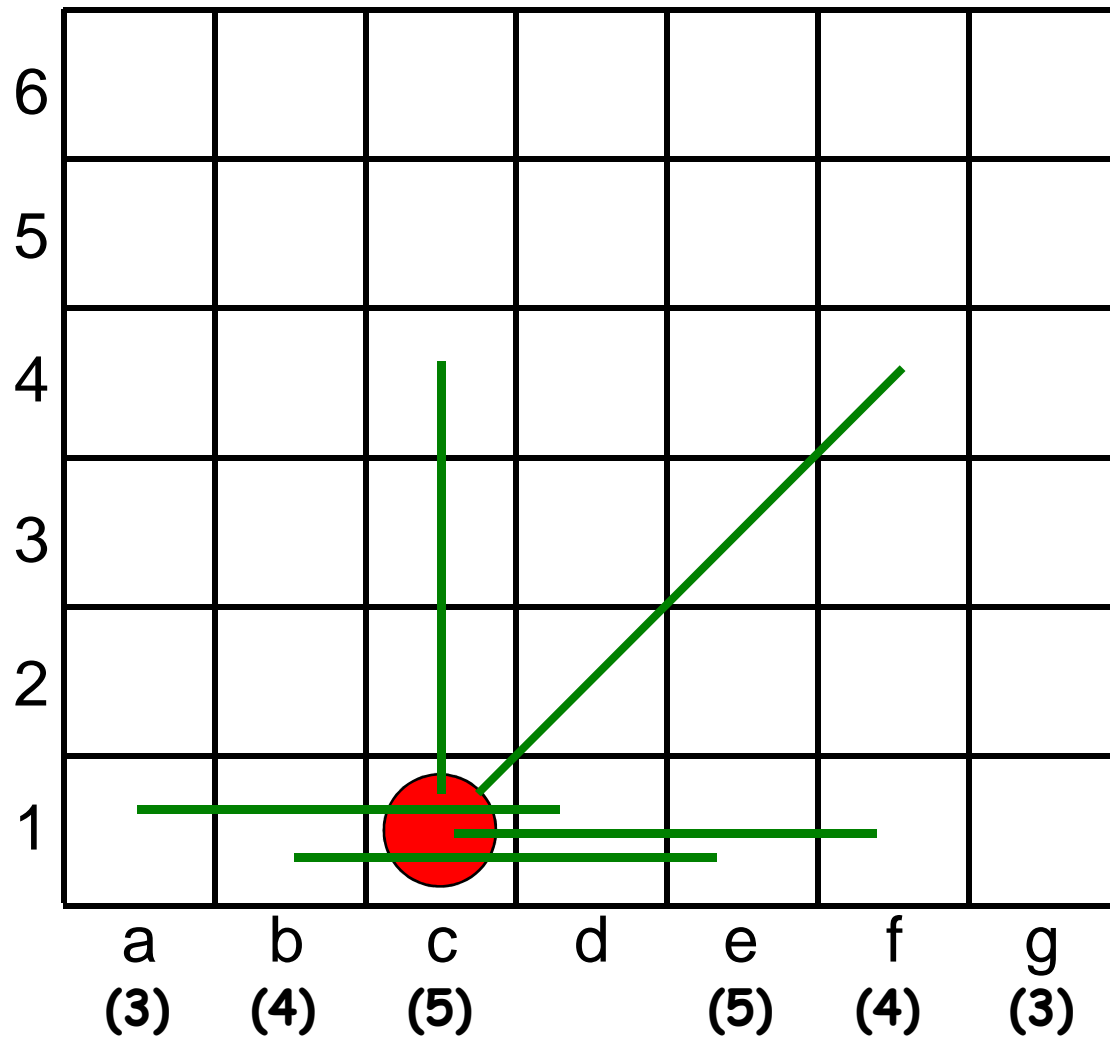
1. Število možnih četvork



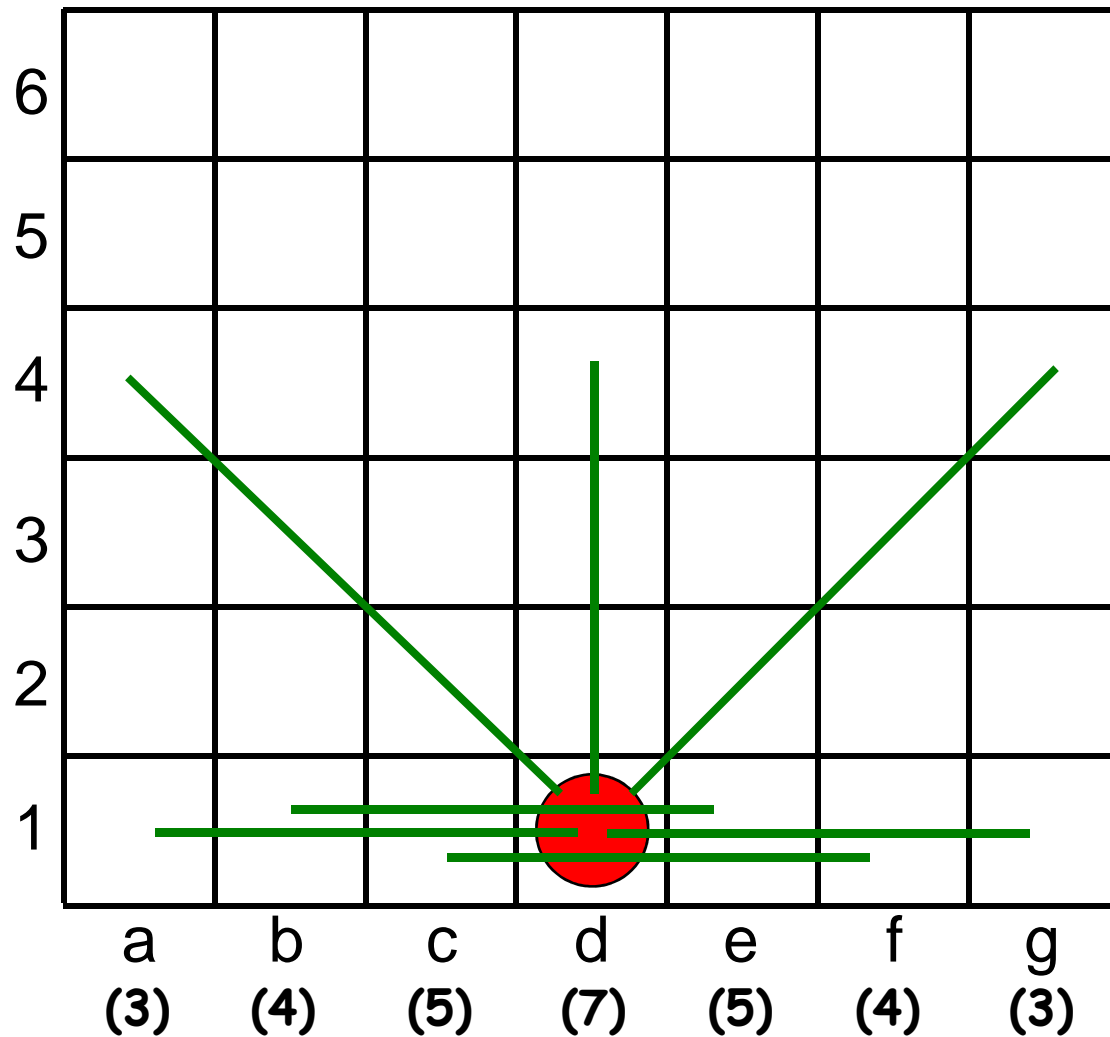
1. Število možnih četvork



1. Števílo možnih četvork

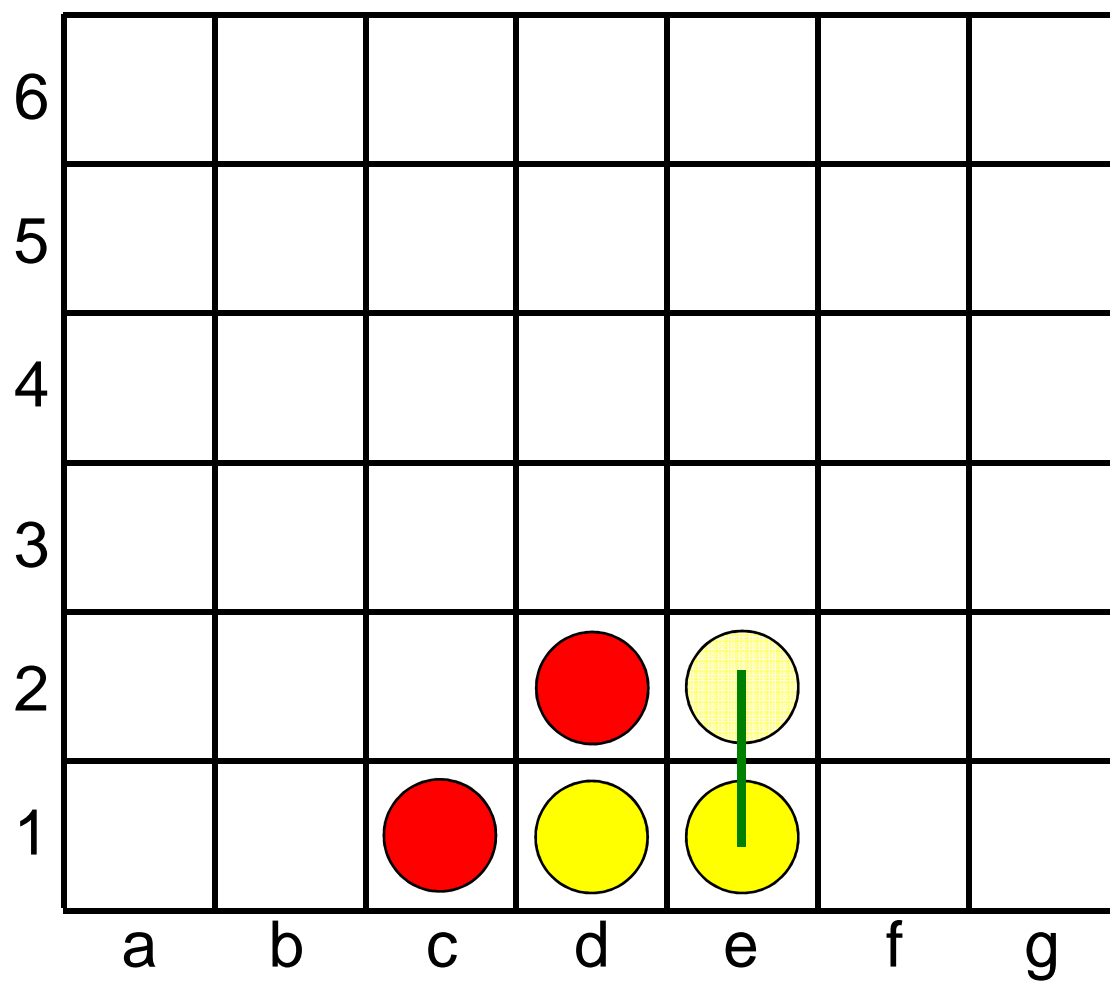


1. Število možnih četvork



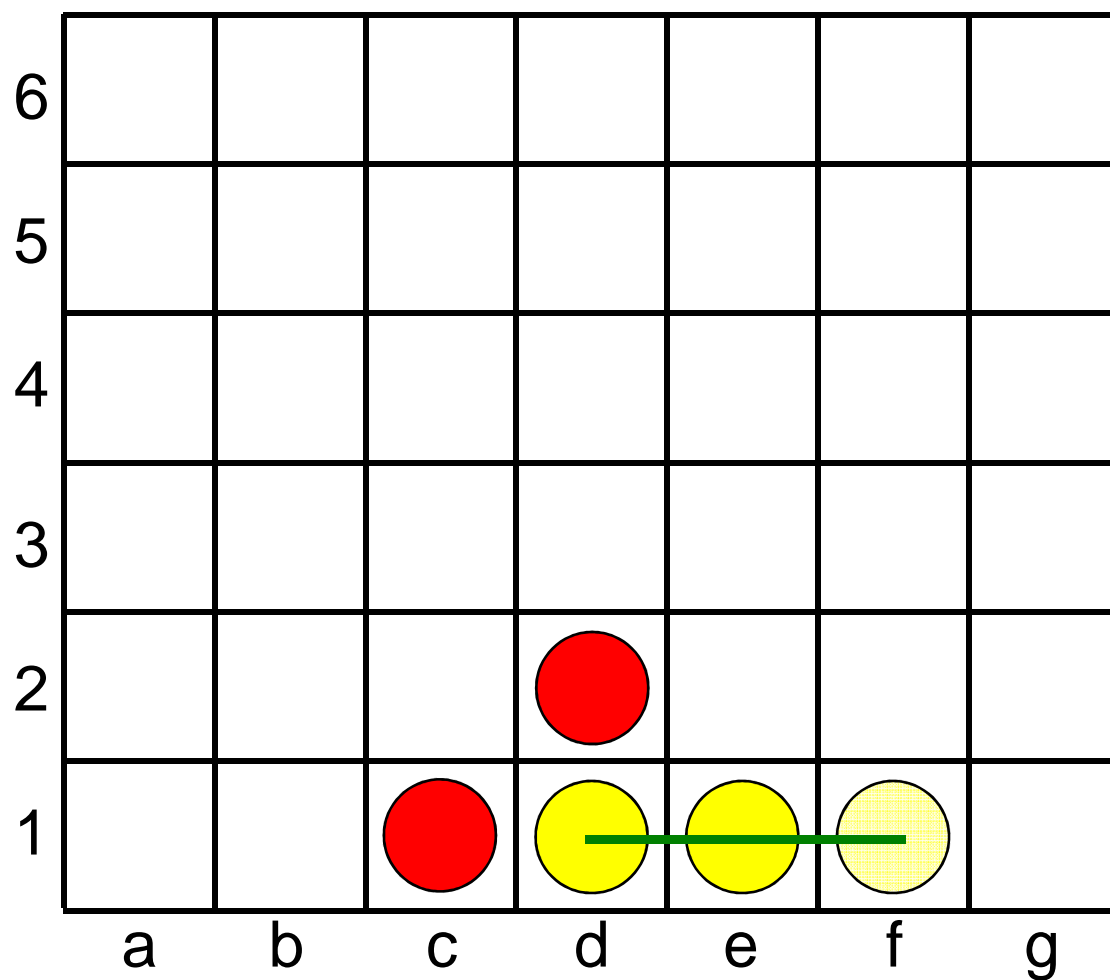
Več kot jih je,
boljše je !!!

2. Trenutno žetonov v vrsti



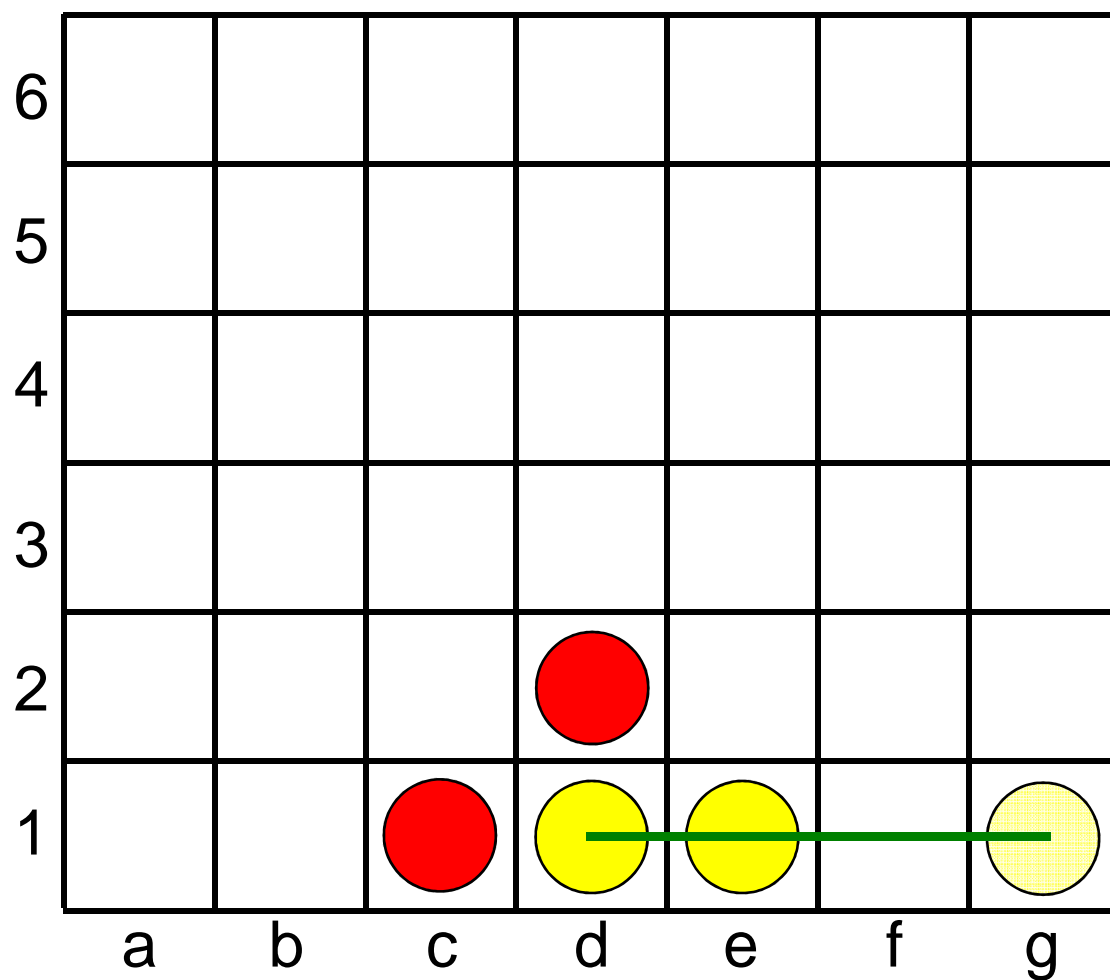
(2)

2. Trenutno žetonov v vrsti



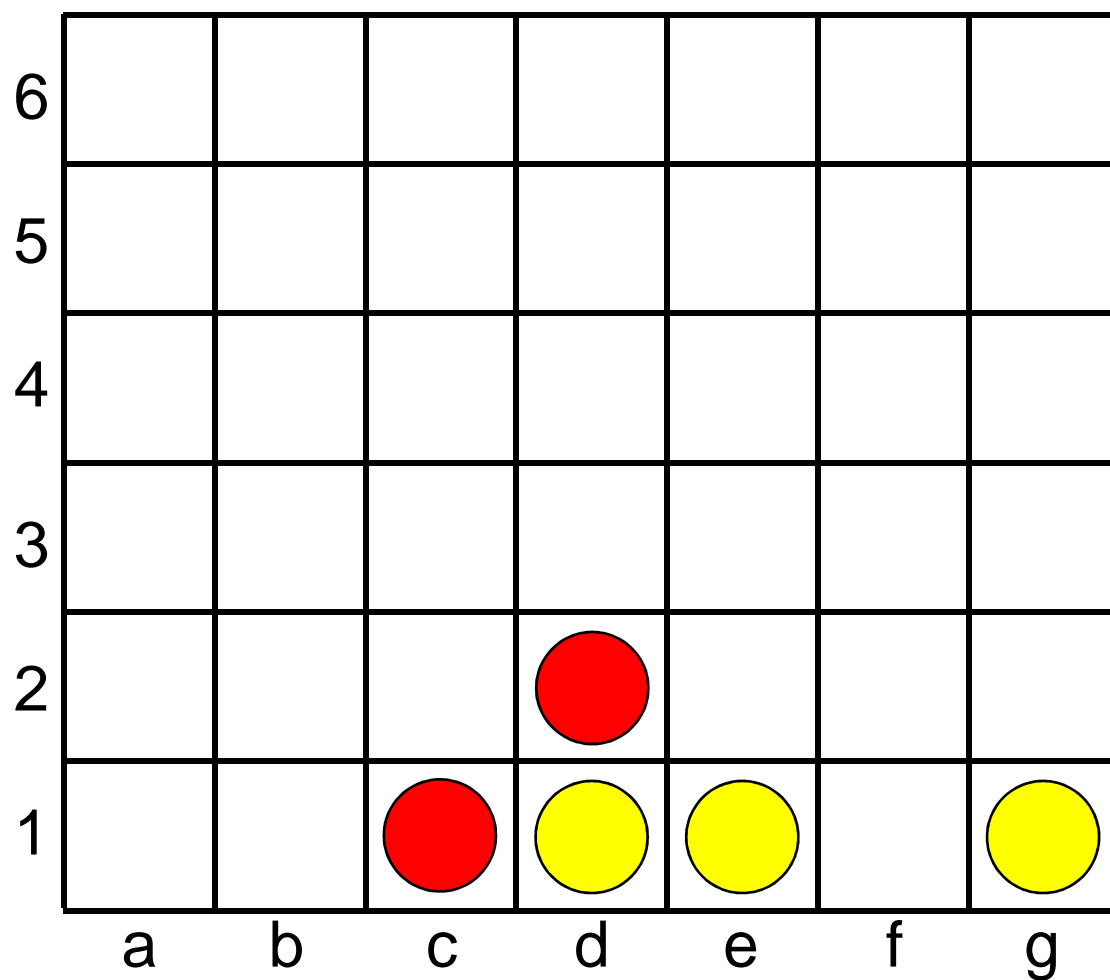
(3)

2. Trenutno žetonov v vrsti



(3)

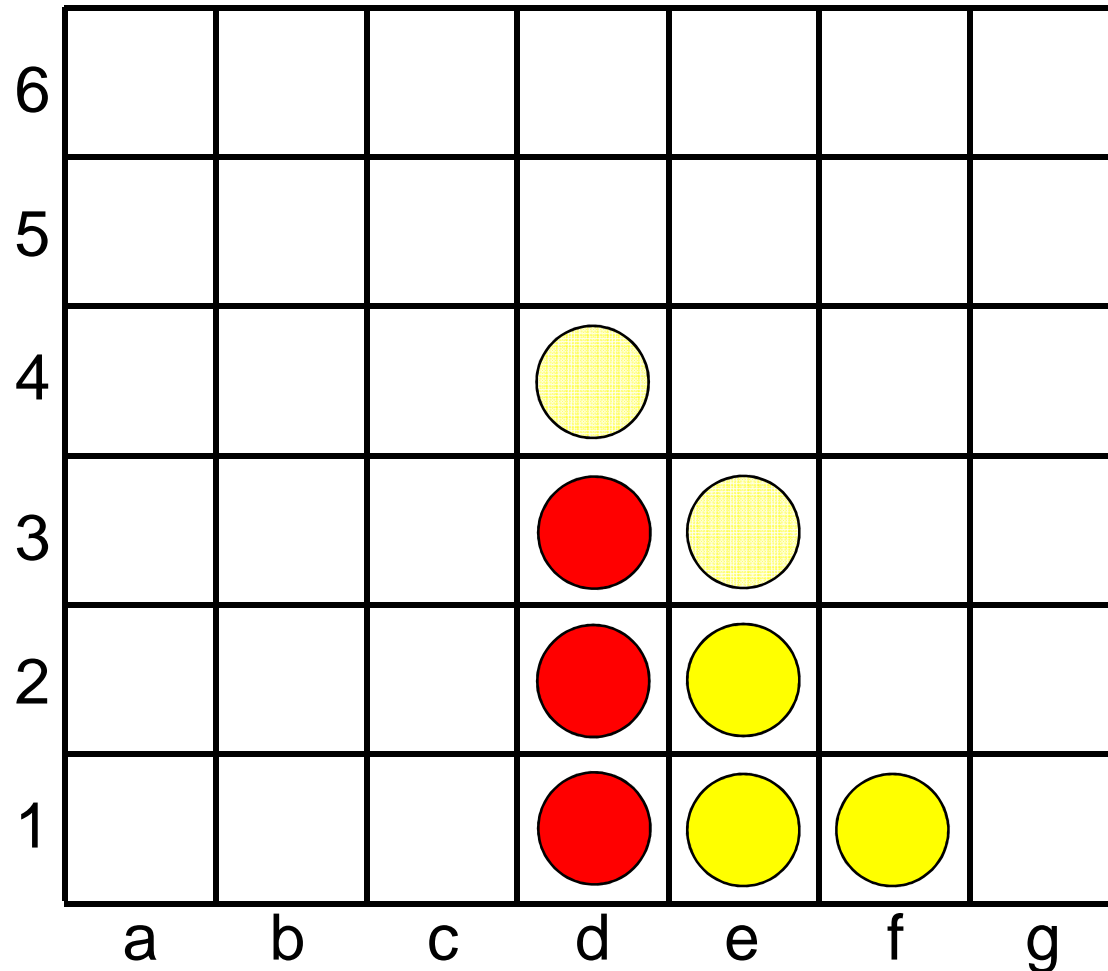
2. Trenutno žetonov v vrsti



Več kot jih je,
boljše je !!!

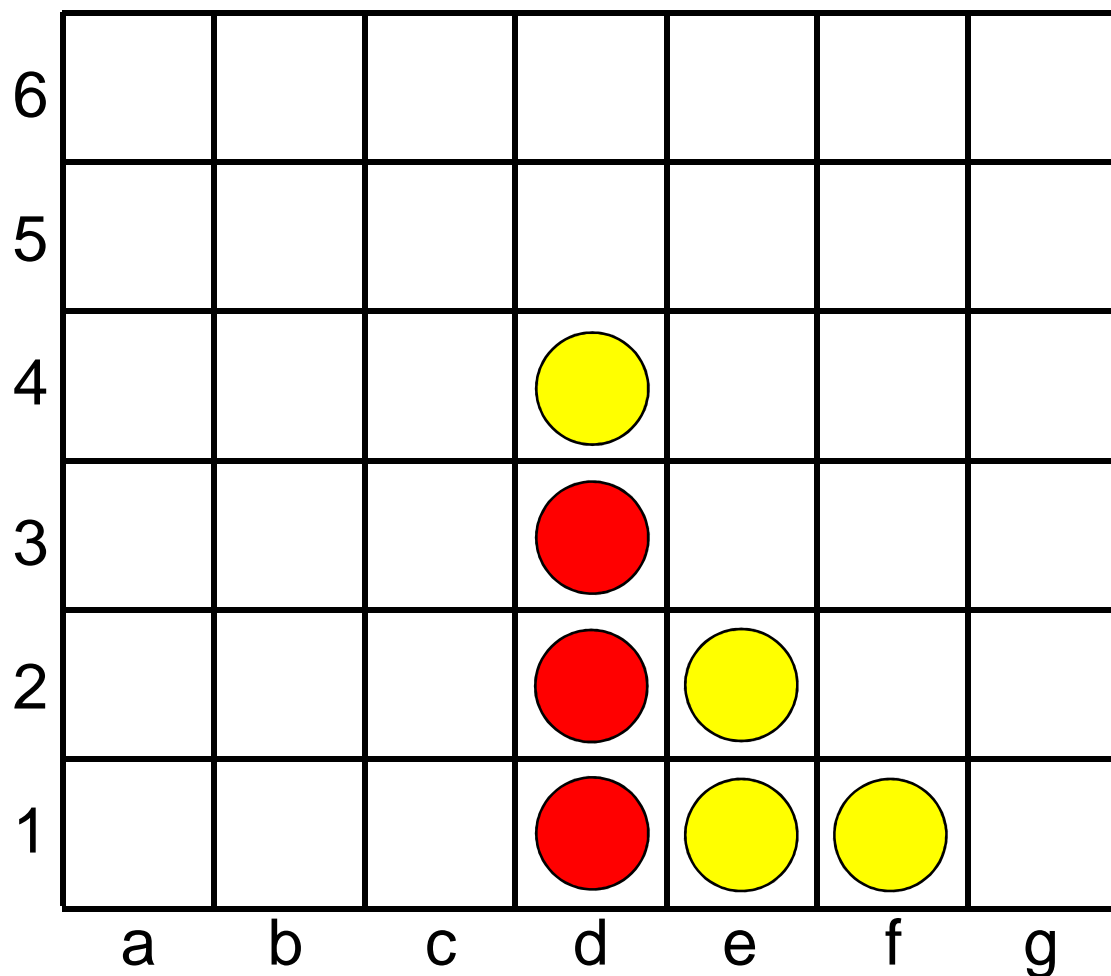
(3)

3. Zapiranje nasprotnikove dobre poteze



Da

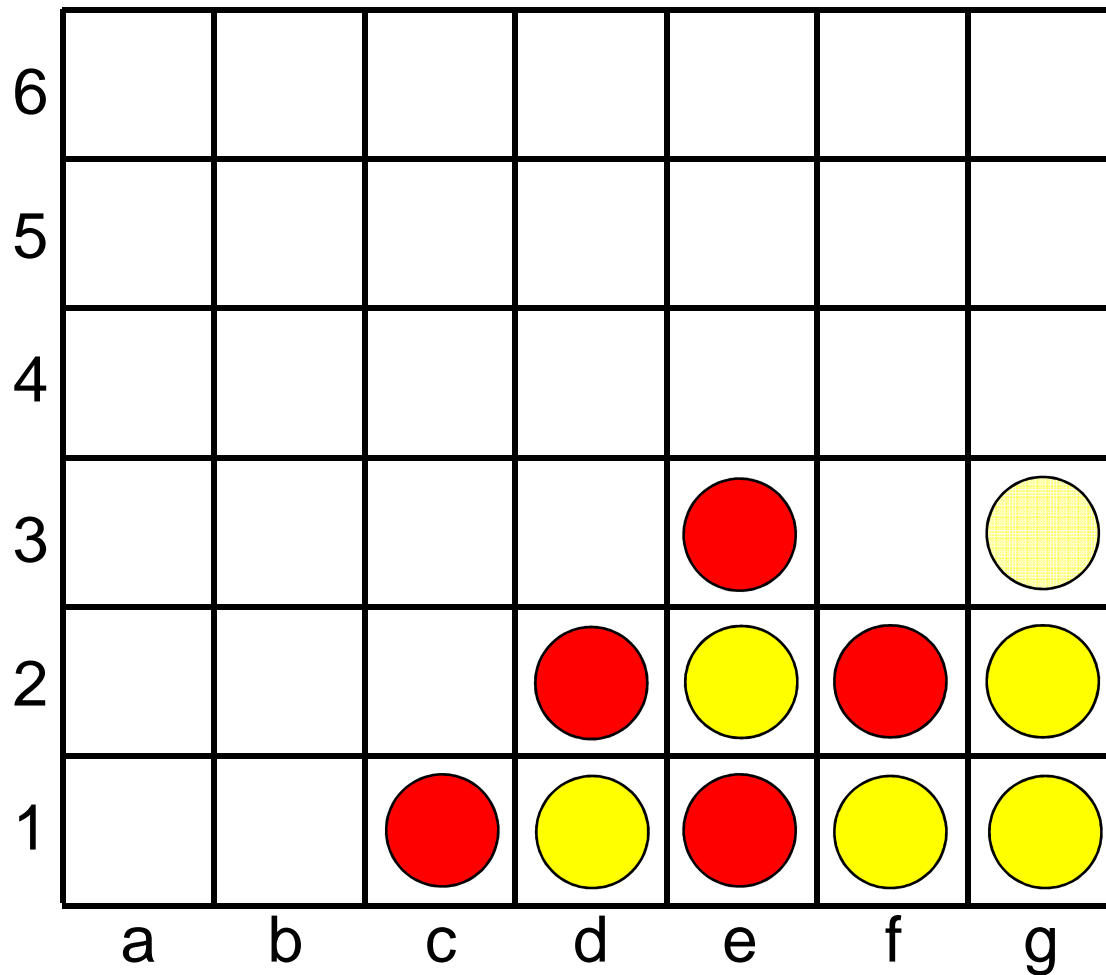
3. Zapiranje nasprotnikove dobre poteze



Da je boljše
kot Ne !!!

Da

4. Prostor zgoraj = prednost



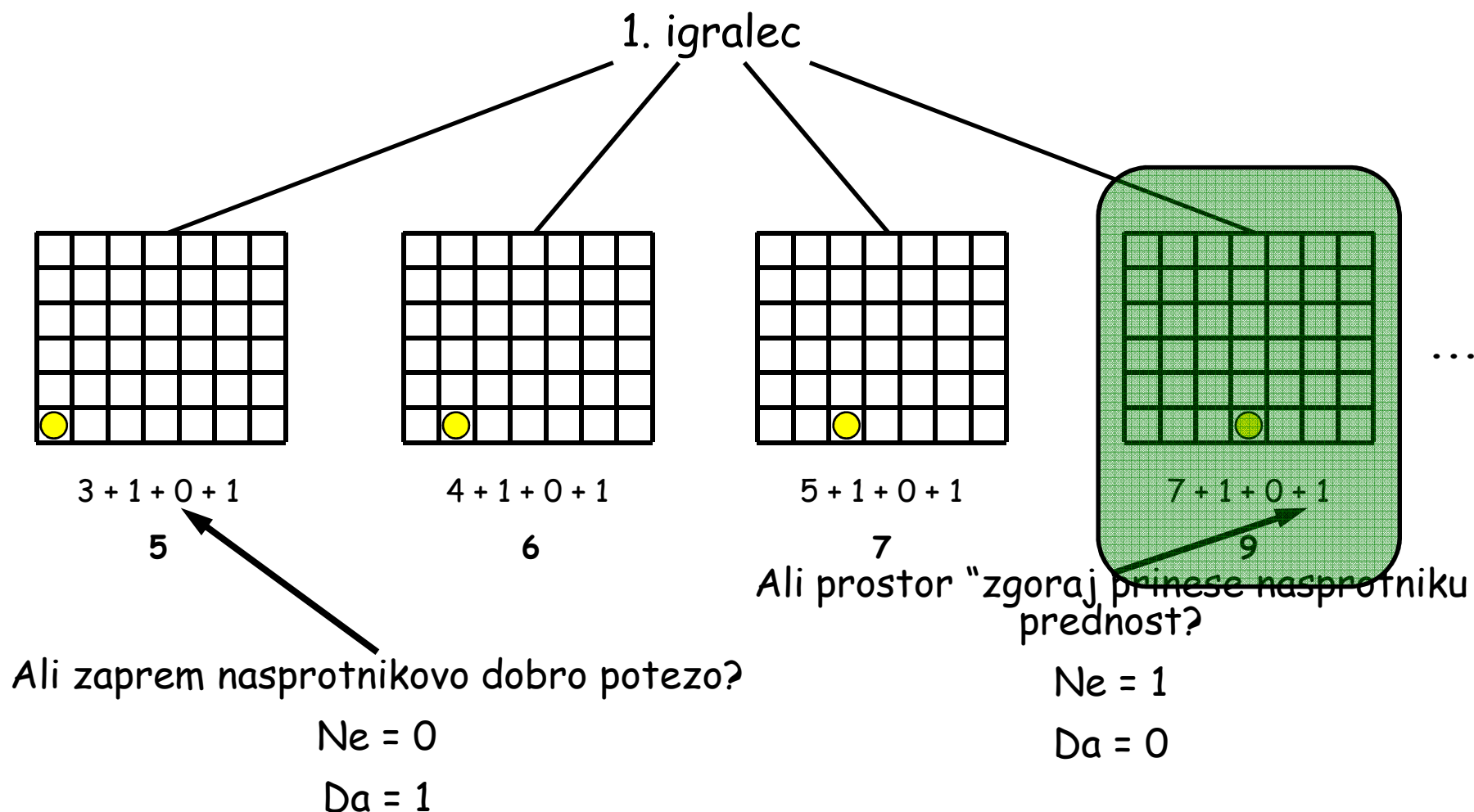
Ne je boljše
kot Da !!!

Ne

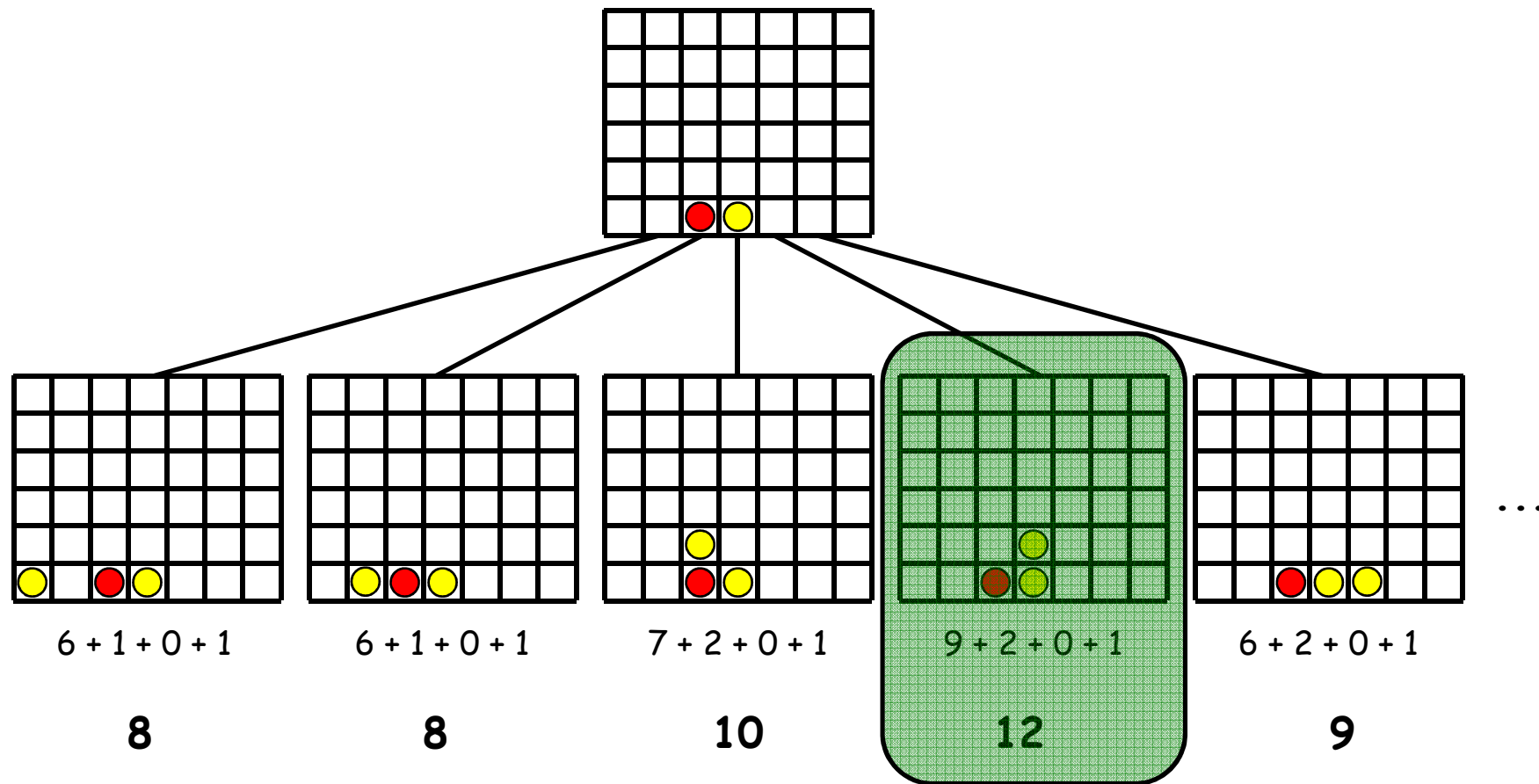
Kako torej zmagamo?

- Če upoštevamo prej opisane 4 strategije
 1. Število možnih četvork (> je boljše)
 2. Trenutno število žetonov v vrsti (> je boljše)
 3. Ali zaprem nasprotnikovo "dobro" potezo?
(Da je boljše)
 4. Ali prostor "zgoraj" prinese nasprotniku prednost? (Ne je boljše)
- Seštejemo vrednosti pod 1., 2., 3. in 4. in izberemo največji seštevek

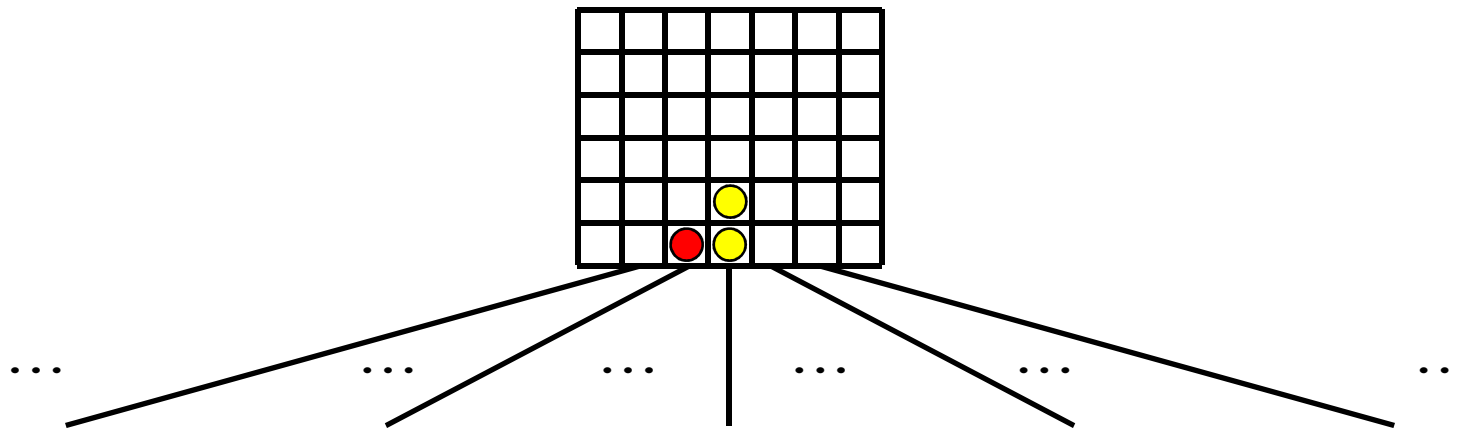
Primer uporabe strategij



Primer uporabe strategij



Primer uporabe strategij



Vloga računalnika

- Omenjene 4 strategije prepisemo v računalniški program ("sprogramiramo")
- Računalnik lahko namesto "nas" igra "4 v vrsto"
- Strategije ne zagotavljajo zmage v vsakem primeru

Kako zmagati igro? (2)

- Z upoštevanjem vseh potez:
 - V trenutni situaciji poskušamo (v mislih / z računalnikom) odigrati **vse** možne kombinacije potez
 - Za vsako možno potezo izračunamo verjetnost, da bomo zmagali
 - Izberemo tisto potezo, ki ima najvišjo verjetnost za zmago
 - Kako to naredimo?

Pogled naprej (look ahead)

Igralec 1

Igralec 1 vidi dve 10-ki na desni
Igralec 1 gre desno
Igralec 2 gre levo
Igralec 1 hoče 10-ko in gre levo
Igralec 2 gre desno
Nauk: Naš soigralec ni neumen!!!

Igralec 2

Igralec 1

Igralec 2

10 2 8 6 7 8 2 3 10 1 4 5 10 3 5 7

Min-Max

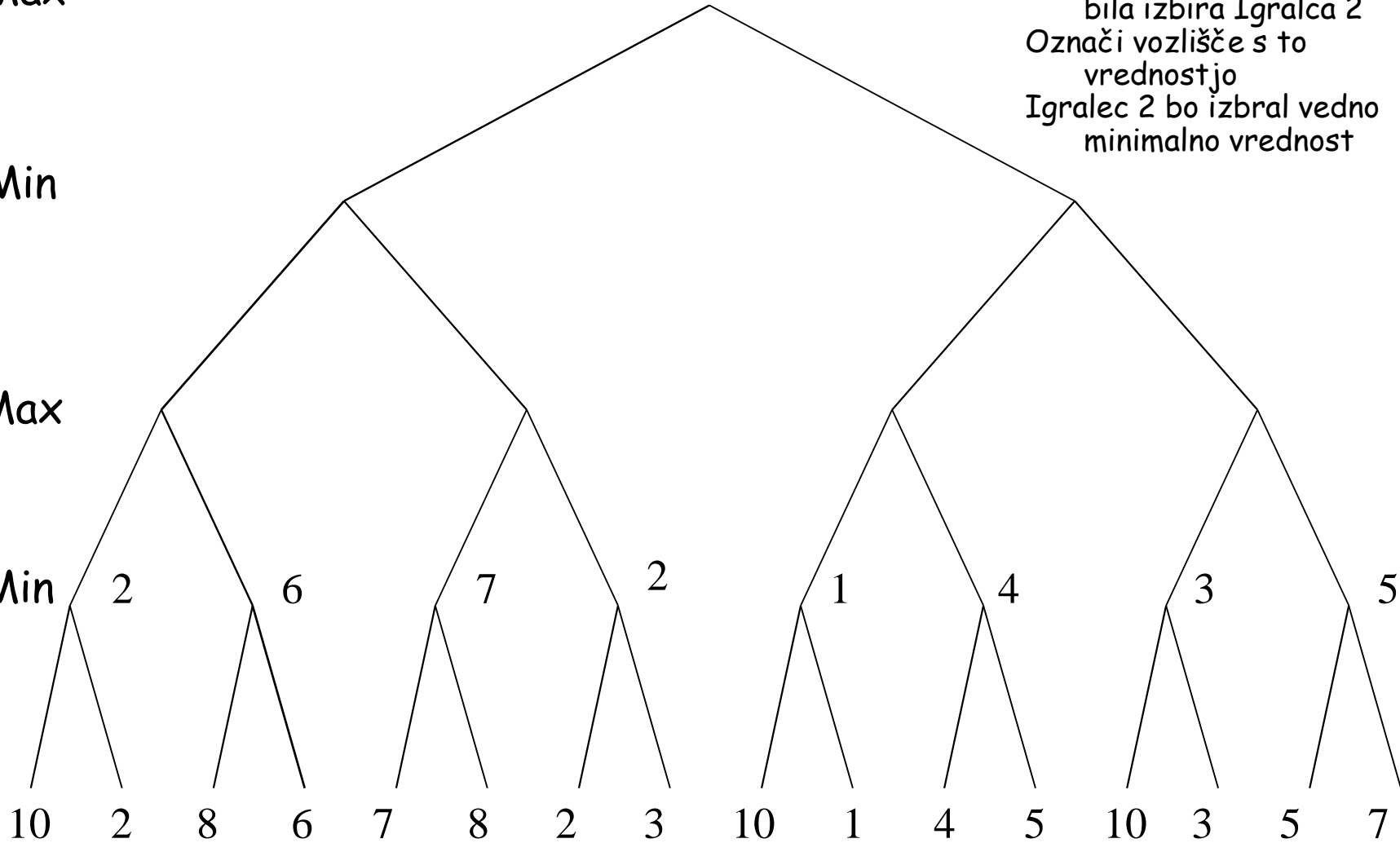
Max

Min

Max

Min

Igralec 1 pogleda kaj bi lahko
bila izbira Igralca 2
Označi vozlišče s to
vrednostjo
Igralec 2 bo izbral vedno
minimalno vrednost



Min-Max

Max

Igralec 1 izbere želeno vrednost
Igralec 2 bo izbral vedno maksimalno vrednost

Min

Max

Min

10

2

8

6

7

8

2

3

10

1

4

5

10

3

5

7

Min-Max

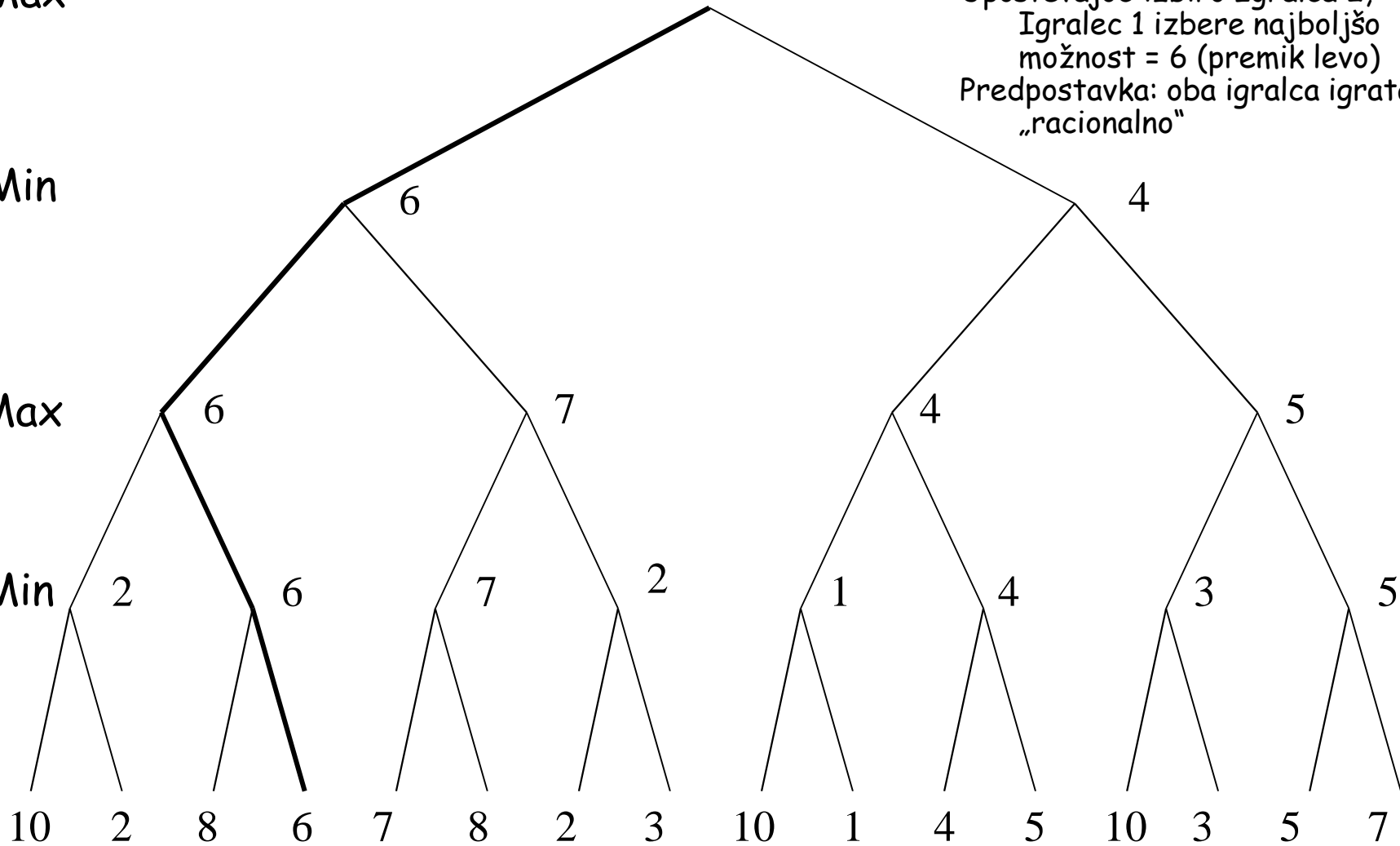
Max

Upoštevajoč izbiro Igralca 2,
Igralec 1 izbere najboljšo
možnost = 6 (premik levo)
Predpostavka: oba igralca igrata
„racionalno“

Min

Max

Min



Min-Max

```
// Node is current board position
// Depth is number of levels in our look-ahead tree
function minimax(node, depth)
    if node is a terminal node or depth = 0
        return the heuristic value of node
    if the adversary is to play at node
        let  $\alpha := +\infty$ 
        foreach child of node
             $\alpha := \min(\alpha, \text{minimax}(\text{child}, \text{depth}-1))$ 
    else // we are to play at node
        let  $\alpha := -\infty$ 
        foreach child of node
             $\alpha := \max(\alpha, \text{minimax}(\text{child}, \text{depth}-1))$ 
    return  $\alpha$ 
```


Min-Max

- Problem:

Zamudno preiskovanje celotnega drevesa možnih rešitev

- Ideja:

Preiskovanje v globino z α - β rezanjem drevesa možnih rešitev

α - β rezanje

Max

Igralec 1 hoče maksimum (10)
Igralec 2 hoče minimum (1)
Igralec 1 je na potezi

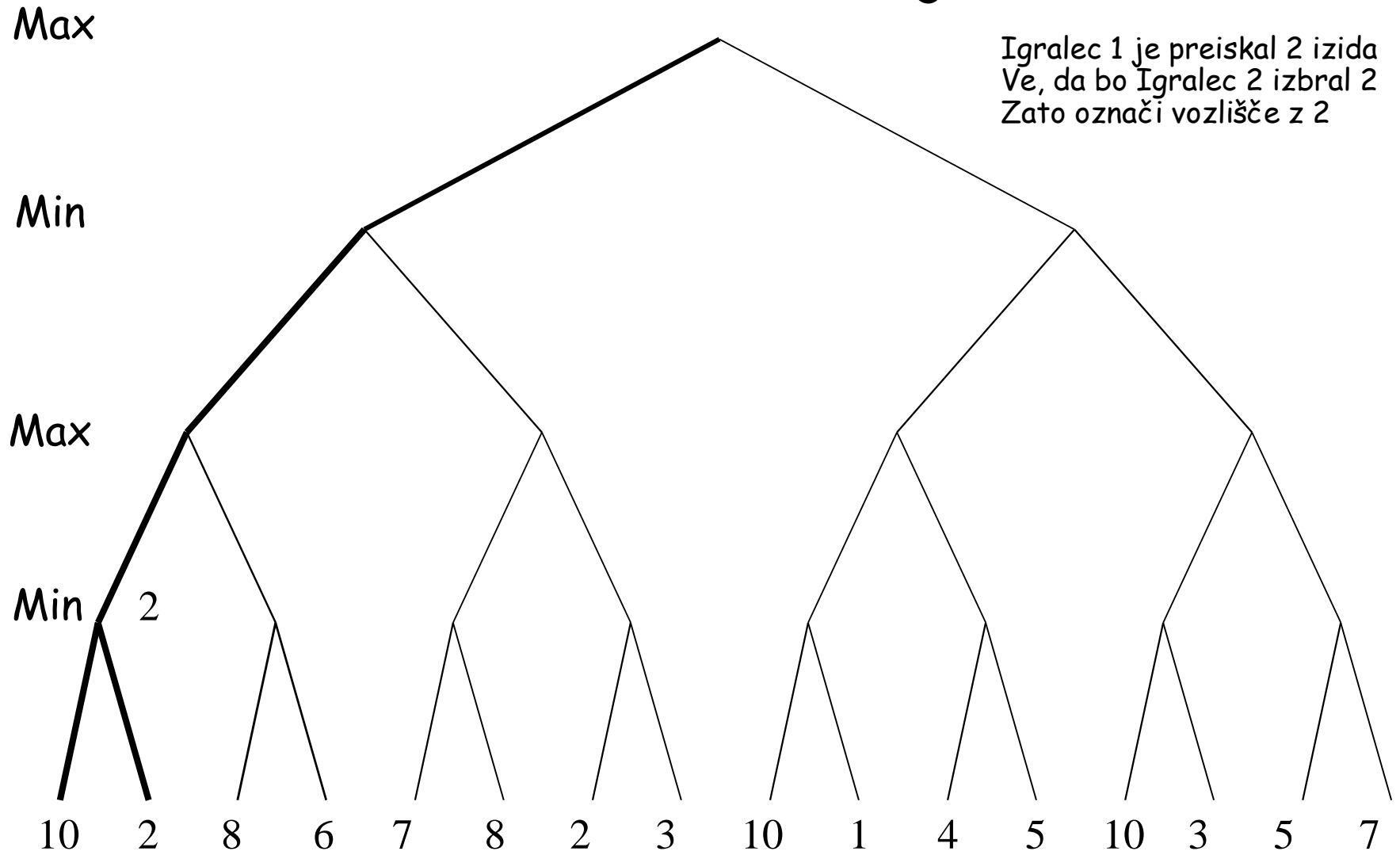
Min

Max

Min

10 2 8 6 7 8 2 3 10 1 4 5 10 3 5 7

α - β rezanje



α - β rezanje

Max

Igralec 1 je preiskal 4 izide
Ve, da bo Igralec 2 izbral 6
(raje kot 8)
Zato ustrezno označi vozlišča

Min

Max

6

Min

2

6

10

2

8

6

7

8

2

3

10

1

4

5

10

3

5

7

α - β rezanje

Max

Igralec 1 je preiskal 6 izidov
Ve, da mu Igralec 2 ne bo dovolil
izbrati 7
Zato mu preostalih 2 vozlišč ni
potrebno preiskati

Min

Max

6

Min

2

6

7

10

2

8

6

7

8

2

3

10

1

4

5

10

3

5

7

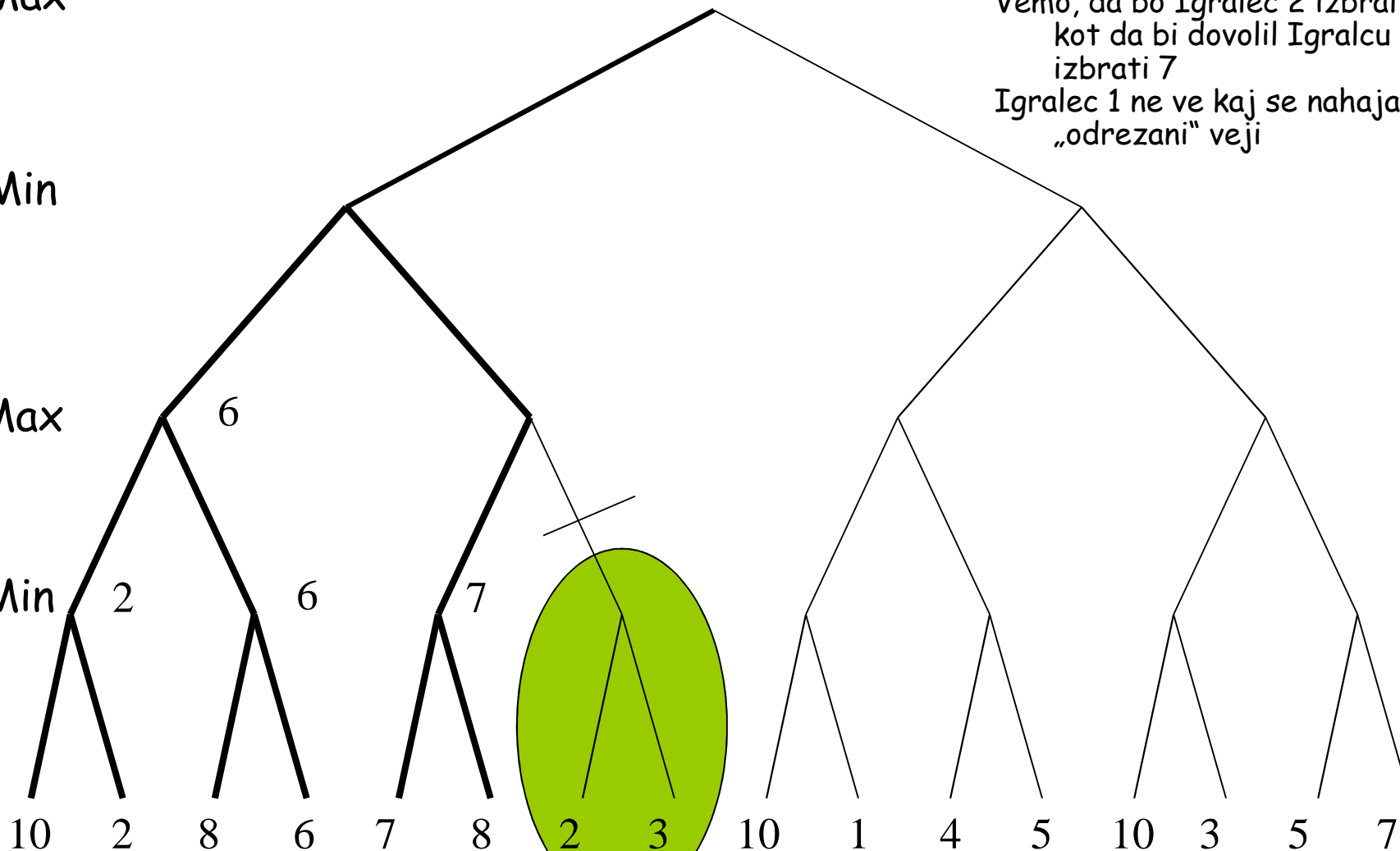
α - β rezanje

Max

Min

Max

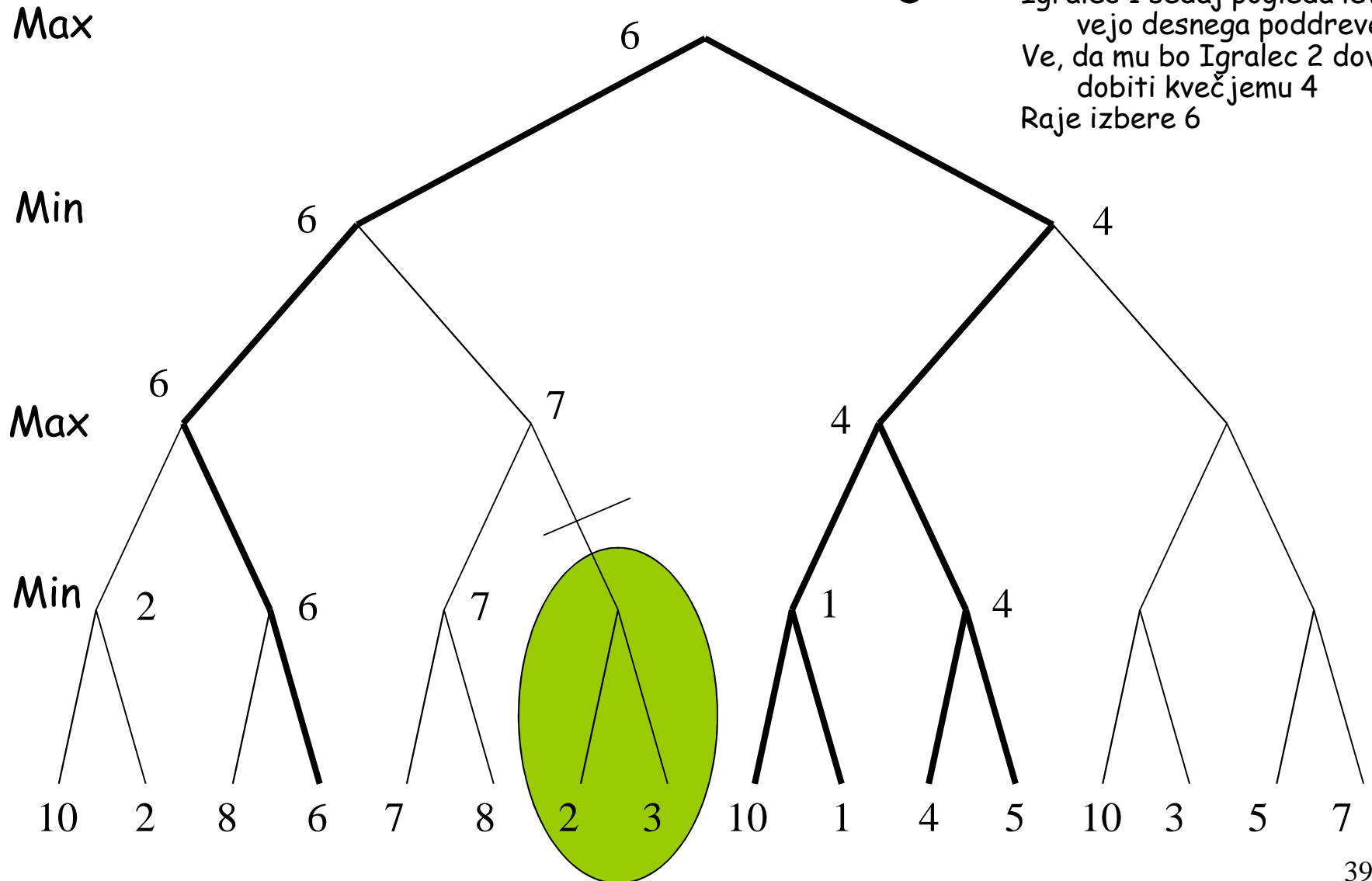
Min



Odrežemo vejo drevesa
Vemo, da bo Igralec 2 izbral 6, raje
kot da bi dovolil Igralcu 1
izbrati 7
Igralec 1 ne ve kaj se nahaja v
„odrezani“ veji

α - β rezanje

Igralec 1 sedaj pogleda levo
vejo desnega poddrevesa
Ve, da mu bo Igralec 2 dovolil
dobiti kvečjemu 4
Raje izbere 6



α - β rezanje

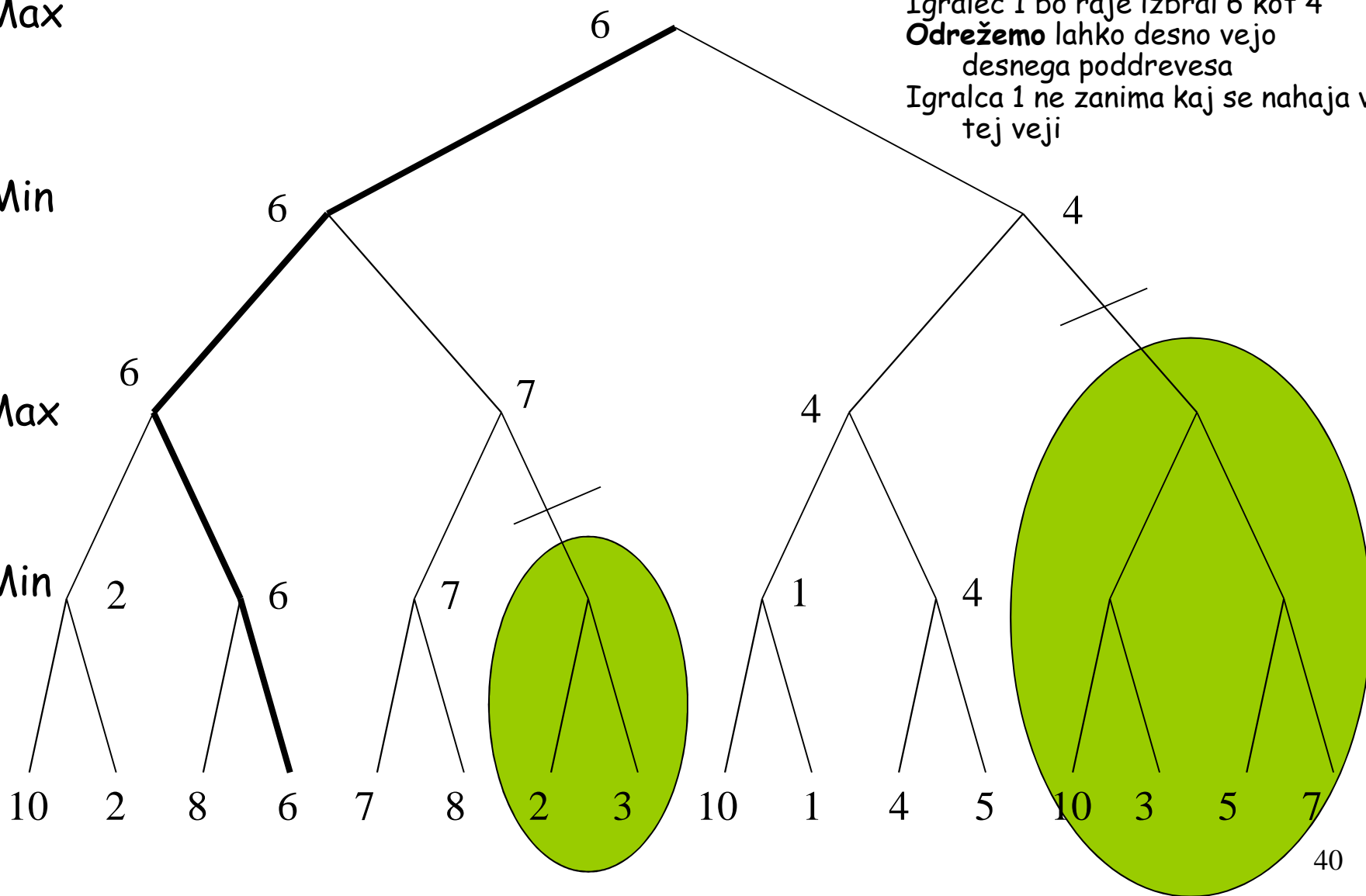
Max

Min

Max

Min

Igralec 1 bo raje izbral 6 kot 4
Odrežemo lahko desno vejo
desnega poddrevesa
Igralca 1 ne zanima kaj se nahaja v
tej veji



α - β rezanje

```
// node = Starting point. Depth = how far to look
function minimax(node, depth)
    return alphabeta(node, depth,  $-\infty$ ,  $+\infty$ )

// Node, depth, value to player, value to opponent
function alphabeta(node, depth,  $\alpha$ ,  $\beta$ )
    if node is a terminal node or depth = 0
        return the heuristic value of node
    foreach child of node
         $\alpha := \max(\alpha, -\text{alphabeta}(\text{child}, \text{depth}-1, -\beta, -\alpha))$ 
        if  $\alpha \geq \beta$ 
            return  $\alpha$ 
    return  $\alpha$ 
```

Koliko je vseh možnosti?

0	1	11	5038572	22	...	33	5611156155990
1	7	12	14164920	23	...	34	6893472773880
2	56	13	35459424	24	...	35	7754703113850
3	252	14	91871208	25	...	36	8385425017200
4	1260	15	214864650	26	...	37	8164755937800
5	4620	16	516936420	27	...	38	7422505398000
6	18480	17	1134183050	28	...	39	5789554210440
7	59815	18	2546423880	29	...	40	3859702806960
8	206780	19	5252058812	30	...	41	1883902560540
9	605934	20	11031780760	31	...	42	538257874440
10	1869840	21	21400686756	32	...	skupaj	70728639995483

Vloga računalnika

- Pomoč pri preiskovanju ogromnega števila možnosti
 - Računalnik lahko najde vse možnosti, a bodisi:
 - Naredi to igranje zelo počasno, bodisi
 - Ne zagotavlja zmage v vsakem primeru (???)
- Pomoč pri oblikovanju boljših strategij igre
 - Računalnik se uči igrati na podlagi že odigranih iger
= **umetna inteligenca**

Primeri računalniške igrice "4 v vrsto"

- Igrica na spletu (on-line):
 - <http://playc4.com/?game=ef1e0f437e821b80ae731845c4792386#> (ANG)
 - <http://www.joyonline.si/index.php?id=igra> (SLO)
- V Javi napisan program za "4 v vrsto":
 - <http://www.ccs.neu.edu/home/eclip5e/classes/csu520/> (ANG)
- "4 v vrsto" na Wikipediji:
 - http://en.wikipedia.org/wiki/Connect_Four (ANG)