

Mesterséges intelligencia

gyakorlat

2. feladat

(backtrack)

Készítette:

Szathmáry László
II. PM.

Feladat: egy általam tetszőlegesen kiválasztott feladat megoldásainak keresése backtrack algoritmus segítségével:

10. Lóugrásban I.

Írjunk rekurzív programot, amely egy adott mezőről indulva a lóugrás szabályainak megfelelően bejár egy 5X5-ös sakktáblát, minden mezőt pontosan egyszer érintve!

I. Állapottér megadása

Az első feladatként elkészített állapottér reprezentációhoz képest annyi a változás, hogy szükségünk lesz egy **x,y** globális változópárra, amelyben azt a pozíciót fogjuk letárolni, hogy a felhasználó hova teszi kezdetben a lovat:

```
#define N 5
enum elem {ures, foglalt, lo};
elem tomb[N][N];
int x,y;
```

Így a kezdőállapot megadása a következőképpen történik:

II. Kezdőállapot megadása

```
void kezdoallapot()
{
    int i,j;

    printf("Feladat: egy %dX%d-os sakktabla bejarasa lougrasban.\n"
           "Adja meg a kezdopoziciot (a szamozas 0-tol kezdodik!):\n\n", N,N);
    printf("Sor (0-%d): ", N-1); scanf("%d", &x);
    printf("Oszlop (0-%d): ", N-1); scanf("%d", &y);

    for (i=0; i<N; ++i)
        for (j=0; j<N; ++j)
            tomb[i][j]=ures;
    tomb[x][y]=lo;
}
```

Most tehát az **x,y** globális változókban tároltuk le a kezdőpozíciót, s a táblán ennek helynek az állapotát **lo**-ra állítottuk.

III. Végállapot megadása

IV. Operátorok

Nincs változás.

A megoldáskeresés során előálló megoldásokhoz vezető lépéseket célszerű veremben tárolni. A verem aljára berakjuk a kezdőpozíciót (ezért volt szükség az **x,y** globális változókra), s erre pakoljuk rá a következő lépéseket. Backtrack, azaz visszalépés esetén a verem tetejéről törölünk egy elemet.

A verem megvalósítására kétirányú láncolt listát alkalmaztam a következőképpen:

```
struct rek {
    int x,y;
    struct rek *elozo, *kov;
} *fej = NULL, *vege = NULL;           //verem inicializálás
```

A rekordnak van 2 adatmezője (x-sor, y-oszlop), ill. két mutató mezője. **fej** a verem aljára, **vege** a verem tetejére mutat. (Mivel legalul van a kezdőpozíció, ezért egy lehetséges megoldás megtalálásakor a verem aljától indulva fogjuk kiíratni a megoldáshoz vezető utat).

Verem esetén két kezelő műveletre van szükség: berak (push), kivesz (pop):

```
void berak(int x, int y)                //a verem tetejére berakja
{
    struct rek *uj = (struct rek *) malloc(sizeof(struct rek));
    if (!uj) hiba();

    uj->x = x; uj->y = y; uj->kov = NULL;

    if (fej==NULL) fej=vege=uj, uj->elozo = uj->kov = NULL;
    else vege->kov = uj, uj->elozo = vege, vege=uj;
}
```

```
void kivesz()                           //a verem tetejéről töröl
{
    if (fej==vege) free(fej), fej=vege=NULL;
    else vege=vege->elozo, free(vege->kov), vege->kov=NULL;
}
```

```
void hiba()
{
    printf("Memoriafoglalasi hiba.\n");
    exit(-1);
}
```

Dinamikus tárkezelés esetén illik leellenőrizni, hogy sikeres volt-e a memóriefoglalás, vagy sem. Ha nem, akkor memóriefoglalási hibával kilépünk a programból.

A **kiir()** függvény akkor hívódik meg, amikor találtunk egy megoldást, vagyis teljesül a végfeltétel, ill. amikor a vezérlő (**bejar()**) befejezte a működését. Ez utóbbi esetben a következő üzenet jelenik meg: "Nem lehet bejarni vagy nincs több megoldas". Ezzel jelezzük, hogy a program befejezte a működését. Ennél a feladatnál ui. előfordulhat olyan eset is, hogy arról a kezdőpozícióról NEM lehet lóugrásban bejarni a táblát.

A főprogram a következőképpen néz ki:

```
void main()
{
    clrscr();
    kezdoallapot();
    bejar(x,y);
    kiir();           //tehát itt is meghívjuk még egyszer
}
```

A `kiir()` függvény:

```
void kiir()
{
    struct rek *akt = fej;           //verem aljáról indulunk
    static long szamlal=0;

    if (!vege_van()) printf("Nem lehet bejarni vagy nincs tobb megoldas.\n");
    else
    {
        printf("%3ld. megoldas:\n", ++szamlal);
        while (akt)
        {
            printf("(%d,%d) ", akt->x, akt->y);
            akt = akt->kov;           //verem teteje felé haladunk
        }
        printf("\n\n");
    }
}
```

A vezérlő részen belül először a **berak** függvényt hívjuk meg, amely elhelyezi a veremben a kezdőpozíciót. (Ezt korábban a felhasználótól kértük be). Ezután megvizsgáljuk, hogy a tábla kielégíti-e a végállapotot. Ha igen, akkor a **kiir** segítségével megjelenítjük ezt a lehetséges megoldást, majd a *return* segítségével visszalépünk a rekurzióban (backtrack). Törlődik az utolsó lépés, amely hatására a végfeltétel igazgá vált. A ló visszalép arra a mezőre, ahonnan ellépett, ezt **lo** állapotúra állítjuk, s ahonnan most visszalépett, az **ures** állapotú lesz. Ezután még a verem tetejéről is ki kell törölni azt a pozíciót, amelyet most állítottunk üresre.

Miután megvolt a visszalépés, meg kell nézni, hogy a lehetséges operátorok közül melyet használhatjuk, melyiknek teljesül az előfeltétele.

Ha zsákutcába jutottunk, vagyis egyetlen operátor sem alkalmazható, akkor ismét visszalépünk a rekurzióban, backtrack történik.

```

void bejar(int x, int y) //ez lesz a vezérlő
{
    berak(x,y);
    if (vege_van())
    {
        kiir();
        return;
    }
    else
    {
        if (balra_le_ef(x,y)) //1.
        {
            balra_le(x,y);
            bejar(x+1, y-2);
            tomb[x+1][y-2]=ures; tomb[x][y]=lo; kivesz(); //backtrack
        }
        if (balra_fel_ef(x,y)) //2.
        {
            balra_fel(x,y);
            bejar(x-1, y-2);
            tomb[x-1][y-2]=ures; tomb[x][y]=lo; kivesz();
        }
        if (fel_balra_ef(x,y)) //3.
        {
            fel_balra(x,y);
            bejar(x-2, y-1);
            tomb[x-2][y-1]=ures; tomb[x][y]=lo; kivesz();
        }
        if (fel_jobbra_ef(x,y)) //4.
        {
            fel_jobbra(x,y);
            bejar(x-2, y+1);
            tomb[x-2][y+1]=ures; tomb[x][y]=lo; kivesz();
        }
        if (jobbra_fel_ef(x,y)) //5.
        {
            jobbra_fel(x,y);
            bejar(x-1, y+2);
            tomb[x-1][y+2]=ures; tomb[x][y]=lo; kivesz();
        }
        if (jobbra_le_ef(x,y)) //6.
        {
            jobbra_le(x,y);
            bejar(x+1, y+2);
            tomb[x+1][y+2]=ures; tomb[x][y]=lo; kivesz();
        }
        if (le_jobbra_ef(x,y)) //7.
        {
            le_jobbra(x,y);
            bejar(x+2, y+1);
            tomb[x+2][y+1]=ures; tomb[x][y]=lo; kivesz();
        }
        if (le_balra_ef(x,y)) //8.
        {
            le_balra(x,y);
            bejar(x+2, y-1);
            tomb[x+2][y-1]=ures; tomb[x][y]=lo; kivesz();
        }
    }
}
}

```

A program futási eredménye (a 25 lehetőségből kettőt kiválasztva):

Amikor a tábla bejárható:

Feladat: egy 5X5-os sakktabla bejárása lóugrásban.

Adja meg a kezdopozíciót (a számolás 0-tól kezdődik!):

Sor (0-4): 2

Oszlop (0-4): 2

1. megoldás:

(2,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (2,1) (0,0)
(1,2) (0,4) (2,3) (4,4) (3,2) (4,0)

2. megoldás:

(2,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (2,1) (4,0)
(3,2) (4,4) (2,3) (0,4) (1,2) (0,0)

3. megoldás:

(2,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (2,3) (0,4)
(1,2) (0,0) (2,1) (4,0) (3,2) (4,4)

4. megoldás:

(2,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (2,3) (4,4)
(3,2) (4,0) (2,1) (0,0) (1,2) (0,4)

5. megoldás:

(2,2) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (2,3) (0,4)
(1,2) (0,0) (2,1) (4,0) (3,2) (4,4)

6. megoldás:

(2,2) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (2,3) (4,4)
(3,2) (4,0) (2,1) (0,0) (1,2) (0,4)

7. megoldás:

(2,2) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,2) (4,0)
(2,1) (0,0) (1,2) (0,4) (2,3) (4,4)

8. megoldás:

(2,2) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,2) (4,4)
(2,3) (0,4) (1,2) (0,0) (2,1) (4,0)

9. megoldás:

(2,2) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,2) (0,0)
(2,1) (4,0) (3,2) (4,4) (2,3) (0,4)

10. megoldás:

(2,2) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,2) (0,4)
(2,3) (4,4) (3,2) (4,0) (2,1) (0,0)

11. megoldás:

(2,2) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (2,3) (0,4)
(1,2) (0,0) (2,1) (4,0) (3,2) (4,4)

12. megoldás:

(2,2) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (2,3) (4,4) (3,2) (4,0) (2,1) (0,0) (1,2) (0,4)

13. megoldás:

(2,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (2,3) (0,4) (1,2) (0,0) (2,1) (4,0) (3,2) (4,4)

14. megoldás:

(2,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (2,3) (4,4) (3,2) (4,0) (2,1) (0,0) (1,2) (0,4)

15. megoldás:

(2,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (2,1) (0,0) (1,2) (0,4) (2,3) (4,4) (3,2) (4,0)

16. megoldás:

(2,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (2,1) (4,0) (3,2) (4,4) (2,3) (0,4) (1,2) (0,0)

17. megoldás:

(2,2) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (1,2) (0,0) (2,1) (4,0) (3,2) (4,4) (2,3) (0,4)

18. megoldás:

(2,2) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (1,2) (0,4) (2,3) (4,4) (3,2) (4,0) (2,1) (0,0)

19. megoldás:

(2,2) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (3,2) (4,0) (2,1) (0,0) (1,2) (0,4) (2,3) (4,4)

20. megoldás:

(2,2) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (3,2) (4,4) (2,3) (0,4) (1,2) (0,0) (2,1) (4,0)

21. megoldás:

(2,2) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (2,1) (0,0) (1,2) (0,4) (2,3) (4,4) (3,2) (4,0)

22. megoldás:

(2,2) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (2,1) (4,0) (3,2) (4,4) (2,3) (0,4) (1,2) (0,0)

23. megoldás:

(2,2) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (3,2) (4,0) (2,1) (0,0) (1,2) (0,4) (2,3) (4,4)

24. megoldás:

(2,2) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (3,2) (4,4) (2,3) (0,4) (1,2) (0,0) (2,1) (4,0)

25. megoldás:

(2,2) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (3,2) (4,0) (2,1) (0,0) (1,2) (0,4) (2,3) (4,4)

26. megoldás:

(2,2) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (3,2) (4,4) (2,3) (0,4) (1,2) (0,0) (2,1) (4,0)

27. megoldas:

(2,2) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (1,2) (0,0)
(2,1) (4,0) (3,2) (4,4) (2,3) (0,4)

28. megoldas:

(2,2) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (1,2) (0,4)
(2,3) (4,4) (3,2) (4,0) (2,1) (0,0)

29. megoldas:

(2,2) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (2,3) (0,4)
(1,2) (0,0) (2,1) (4,0) (3,2) (4,4)

30. megoldas:

(2,2) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (2,3) (4,4)
(3,2) (4,0) (2,1) (0,0) (1,2) (0,4)

31. megoldas:

(2,2) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (3,2) (4,0)
(2,1) (0,0) (1,2) (0,4) (2,3) (4,4)

32. megoldas:

(2,2) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (3,2) (4,4)
(2,3) (0,4) (1,2) (0,0) (2,1) (4,0)

33. megoldas:

(2,2) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (2,1) (0,0)
(1,2) (0,4) (2,3) (4,4) (3,2) (4,0)

34. megoldas:

(2,2) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (2,1) (4,0)
(3,2) (4,4) (2,3) (0,4) (1,2) (0,0)

35. megoldas:

(2,2) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,2) (0,0)
(2,1) (4,0) (3,2) (4,4) (2,3) (0,4)

36. megoldas:

(2,2) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,2) (0,4)
(2,3) (4,4) (3,2) (4,0) (2,1) (0,0)

37. megoldas:

(2,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (2,3) (0,4)
(1,2) (0,0) (2,1) (4,0) (3,2) (4,4)

38. megoldas:

(2,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (2,3) (4,4)
(3,2) (4,0) (2,1) (0,0) (1,2) (0,4)

39. megoldas:

(2,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (2,1) (0,0)
(1,2) (0,4) (2,3) (4,4) (3,2) (4,0)

40. megoldas:

(2,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (2,1) (4,0)
(3,2) (4,4) (2,3) (0,4) (1,2) (0,0)

41. megoldas:

(2,2) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (2,1) (0,0)
(1,2) (0,4) (2,3) (4,4) (3,2) (4,0)

42. megoldas:

(2,2) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (2,1) (4,0)
(3,2) (4,4) (2,3) (0,4) (1,2) (0,0)

43. megoldas:

(2,2) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,2) (4,0)
(2,1) (0,0) (1,2) (0,4) (2,3) (4,4)

44. megoldas:

(2,2) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,2) (4,4)
(2,3) (0,4) (1,2) (0,0) (2,1) (4,0)

45. megoldas:

(2,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (2,1) (0,0)
(1,2) (0,4) (2,3) (4,4) (3,2) (4,0)

46. megoldas:

(2,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (2,1) (4,0)
(3,2) (4,4) (2,3) (0,4) (1,2) (0,0)

47. megoldas:

(2,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (2,3) (0,4)
(1,2) (0,0) (2,1) (4,0) (3,2) (4,4)

48. megoldas:

(2,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (2,3) (4,4)
(3,2) (4,0) (2,1) (0,0) (1,2) (0,4)

49. megoldas:

(2,2) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (3,2) (4,0)
(2,1) (0,0) (1,2) (0,4) (2,3) (4,4)

50. megoldas:

(2,2) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (3,2) (4,4)
(2,3) (0,4) (1,2) (0,0) (2,1) (4,0)

51. megoldas:

(2,2) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (1,2) (0,0)
(2,1) (4,0) (3,2) (4,4) (2,3) (0,4)

52. megoldas:

(2,2) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (1,2) (0,4)
(2,3) (4,4) (3,2) (4,0) (2,1) (0,0)

53. megoldas:

(2,2) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (1,2) (0,0)
(2,1) (4,0) (3,2) (4,4) (2,3) (0,4)

54. megoldas:

(2,2) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (1,2) (0,4)
(2,3) (4,4) (3,2) (4,0) (2,1) (0,0)

55. megoldas:

(2,2) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (2,3) (0,4)
(1,2) (0,0) (2,1) (4,0) (3,2) (4,4)

56. megoldas:

(2,2) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (2,3) (4,4)
(3,2) (4,0) (2,1) (0,0) (1,2) (0,4)

57. megoldas:

(2,2) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (2,1) (0,0)
(1,2) (0,4) (2,3) (4,4) (3,2) (4,0)

58. megoldas:

(2,2) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (2,1) (4,0)
(3,2) (4,4) (2,3) (0,4) (1,2) (0,0)

59. megoldas:

(2,2) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (1,2) (0,0)
(2,1) (4,0) (3,2) (4,4) (2,3) (0,4)

60. megoldas:

(2,2) (4,1) (2,0) (0,1) (1,3) (3,4) (4,2) (3,0) (1,1) (0,3) (2,4) (4,3) (3,1) (1,0) (0,2) (1,4) (3,3) (1,2) (0,4)
(2,3) (4,4) (3,2) (4,0) (2,1) (0,0)

61. megoldas:

(2,2) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (1,2) (0,0)
(2,1) (4,0) (3,2) (4,4) (2,3) (0,4)

62. megoldas:

(2,2) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (1,2) (0,4)
(2,3) (4,4) (3,2) (4,0) (2,1) (0,0)

63. megoldas:

(2,2) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (3,2) (4,0)
(2,1) (0,0) (1,2) (0,4) (2,3) (4,4)

64. megoldas:

(2,2) (4,1) (3,3) (1,4) (0,2) (1,0) (3,1) (4,3) (2,4) (0,3) (1,1) (3,0) (4,2) (3,4) (1,3) (0,1) (2,0) (3,2) (4,4)
(2,3) (0,4) (1,2) (0,0) (2,1) (4,0)

Nem lehet bejarni vagy nincs tobb megoldas.

Amikor a tábla nem járható be:

Feladat: egy 5X5-os sakktábla bejárása lóugrásban.

Adja meg a kezdopozíciót (a számozás 0-tól kezdődik!):

Sor (0-4): 0

Oszlop (0-4): 1

Nem lehet bejarni vagy nincs tobb megoldas.