

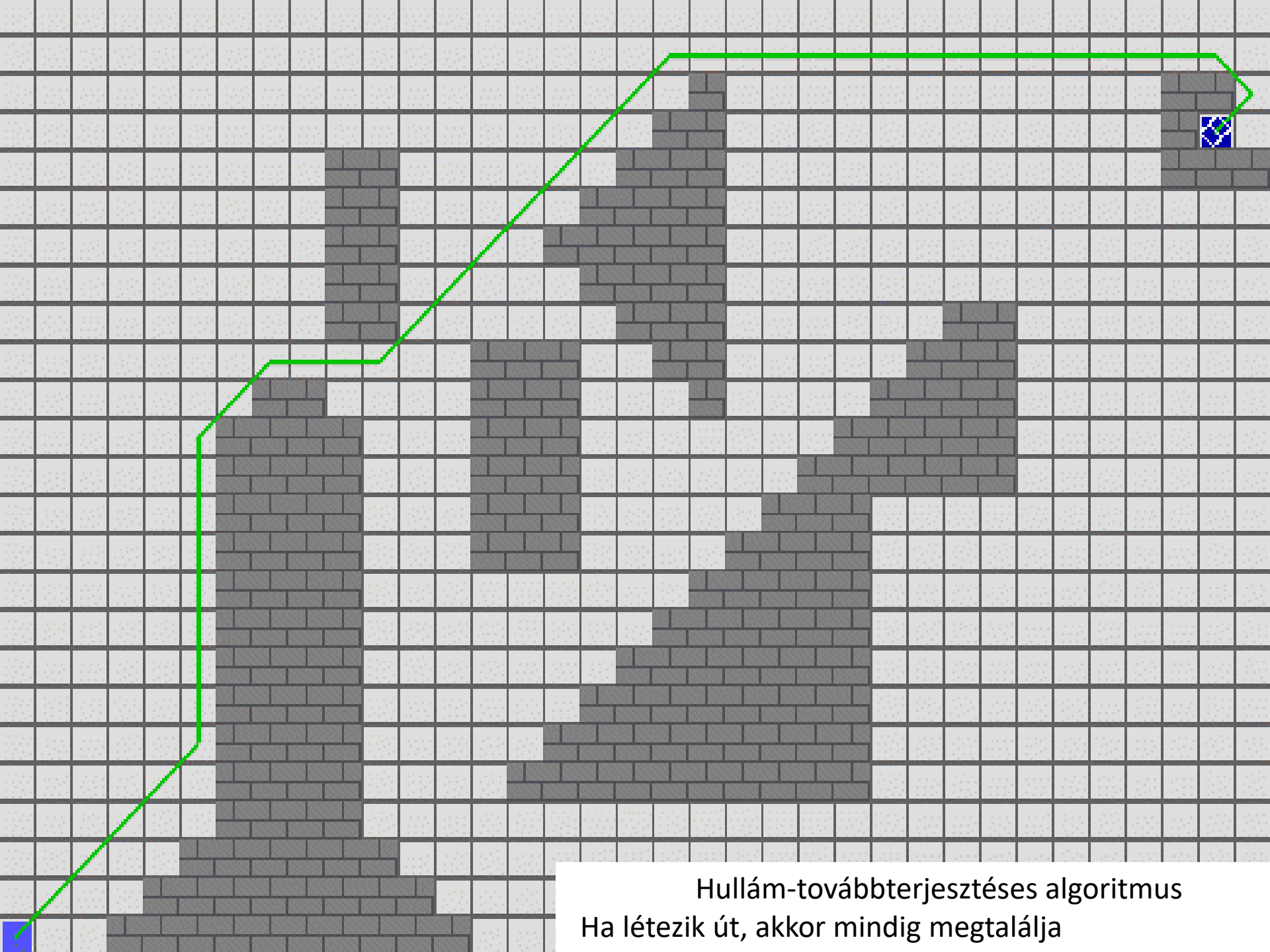
Intelligens Rendszerek

Hullám-továbbterjesztésen alapuló navigáció

Navigáció

Akadályelkerülés és pályatervezés

- szabályalapú algoritmus
 - módosított szabályalapú algoritmus
 - neurális-elvű algoritmus
 - tapasztalat szerzésen alapuló algoritmus (öntanuló)
- hullám-továbbterjesztéses algoritmus
 - módosított hullám-továbbterjesztéses algoritmus
- GVD-elvű, gráfbejáráson alapuló algoritmus



Hullám-továbbterjesztéses algoritmus
Ha létezik út, akkor mindig megtalálja

Matlab keretprogram

- Keretprogram indításkor rákérdez a szabálytábla törlésére (első indításnál mindig üres)
- Alaptérkép betöltése (32d.png)
 - Piros (255,0,0) cella: Robot
 - Zöld (0,255,0) cella: Cél
 - Fehér cellák (255,255,255): üres mező
 - Fekete cellák (0,0,0): akadály
- A keretprogram átalakítja ezt a képet (tömböt) egy 2D tömbbé, melyben a következő értékek szerepelnek:
`emptyVal=0;`
`finishVal=-1;`
`robotVal=-2;`
`obstacleVal=-3;`
Hullámfront cellák = pozitív értékű cellák

Matlab keretprogram

```
newWavefrontCells=1;
```

```
while(newWavefrontCells>0
```

```
%Addig fut a ciklus, amíg van olyan üres cella, ahova a hullám eljuthat
```

```
%Hogyan lehet megoldani, hogy akkor is leálljon a terjesztés, ha a hullám eléri a robotot?
```

```
newWavefrontCells=0;
```

```
    cloneMap=map;
```

```
...
```

```
% 1. feladat
```

```
% Hullámfrontok terjesztése
```

```
% - Minden egyes ciklusban egyetlen új hullámfrontot kell lekezelni
```

```
% - Olyan üres cellákat kell keresni, melynek szomszédja hullámfront
```

```
% cella.
```

```
% - Függőleges, vízszintes irányban +2, átlós irányba +3 értékkel
```

```
% növeljük a hullámfront értékét az vizsgált üres cellákba. Törekedni
```

```
% kell a minimális költségű lépésre.
```

```
% - Le kell lezelni az első lépést, ahol a cél (-1 tartalmú) cellát
```

```
% kell körbevenni hullámfronttal
```

```
% - Figyelni kell a hullámfrontba bevont cellák értékének növelésére(newWavefrontCells)!
```

Matlab keretprogram

- Hullámfront terjesztés után megvizsgálja a keretprogram, hogy a Robot környezetében (8 cella) van e hullámfront.
- Ha igen, akkor van út
- Jöhet a legrövidebb út megkeresése 😊

Matlab keretprogram

- Hullámfront terjesztés után megvizsgálja a keretprogram, hogy a Robot környezetében (8 cella) van e hullámfront.
- Ha igen, akkor van út
- Jöhet a legrövidebb út megkeresése 😊

Matlab keretprogram

```
if (pathavailable)
    globalMax=max(map(:));
    if localMax>0
        pathcomplete=0;
        robotPos(1)=startPos(1);
        robotPos(2)=startPos(2);
        while(~pathcomplete)

% 2. feladat
% Legrövidebb út keresése
% A robot (robotPos) aktuális környezetét meg kell vizsgálni (8
% szomszédos cella) és a legkisebb hullámfront értékűt kell választani (minWave)
% Magát a lépést és útvonal rajzolást a keretprogram elvégzi!

%%% IDE TESSÉK ÍRNI
```

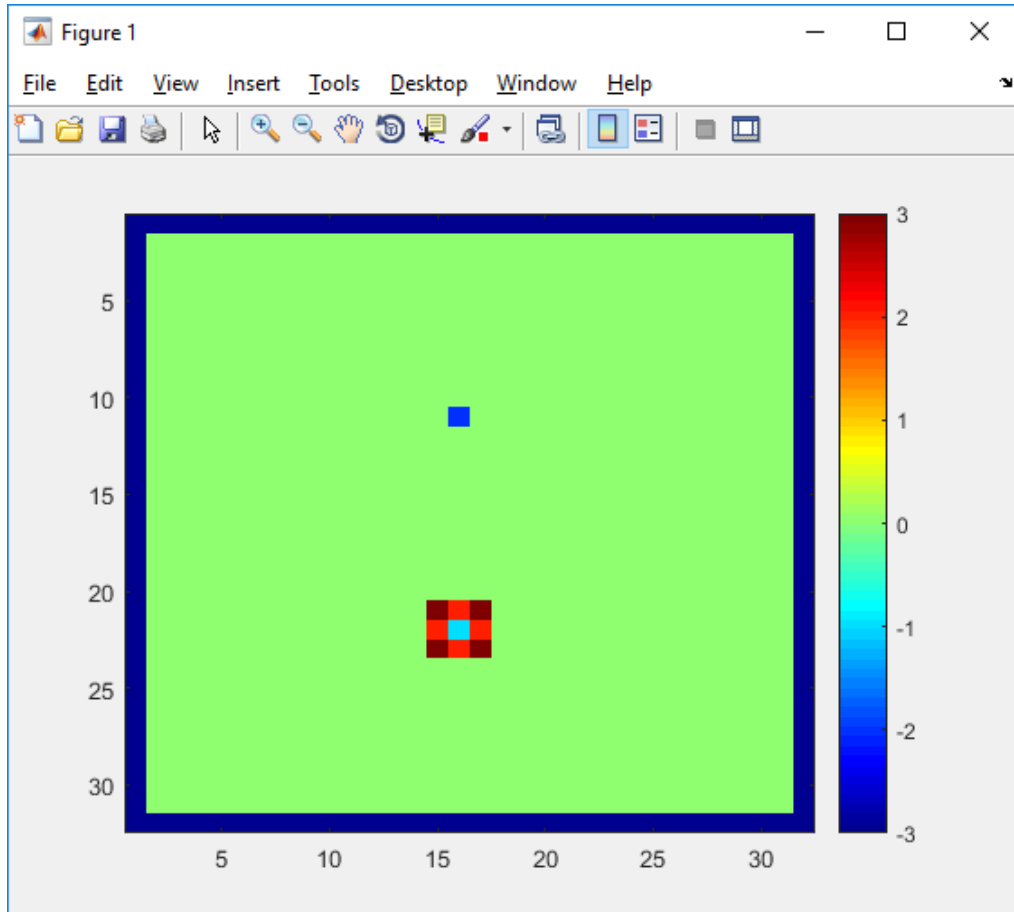
Kezdőállapot – Cél: (-1)

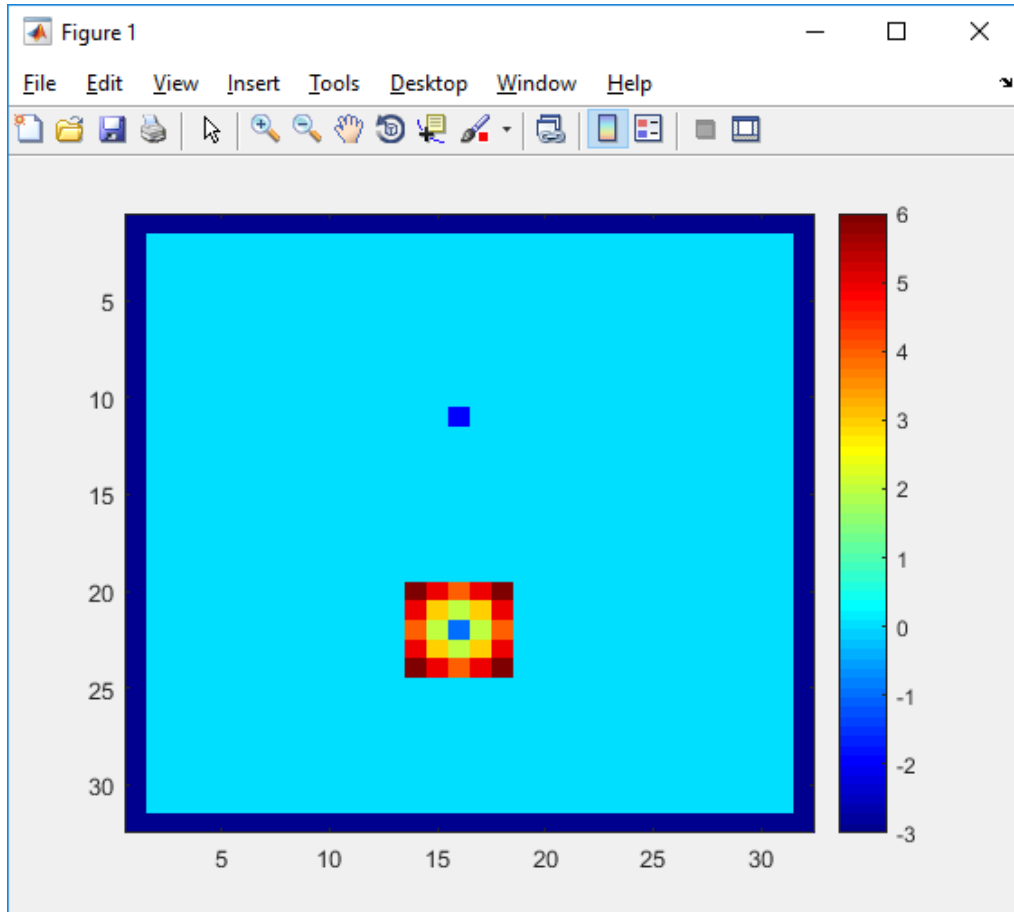
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	-1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

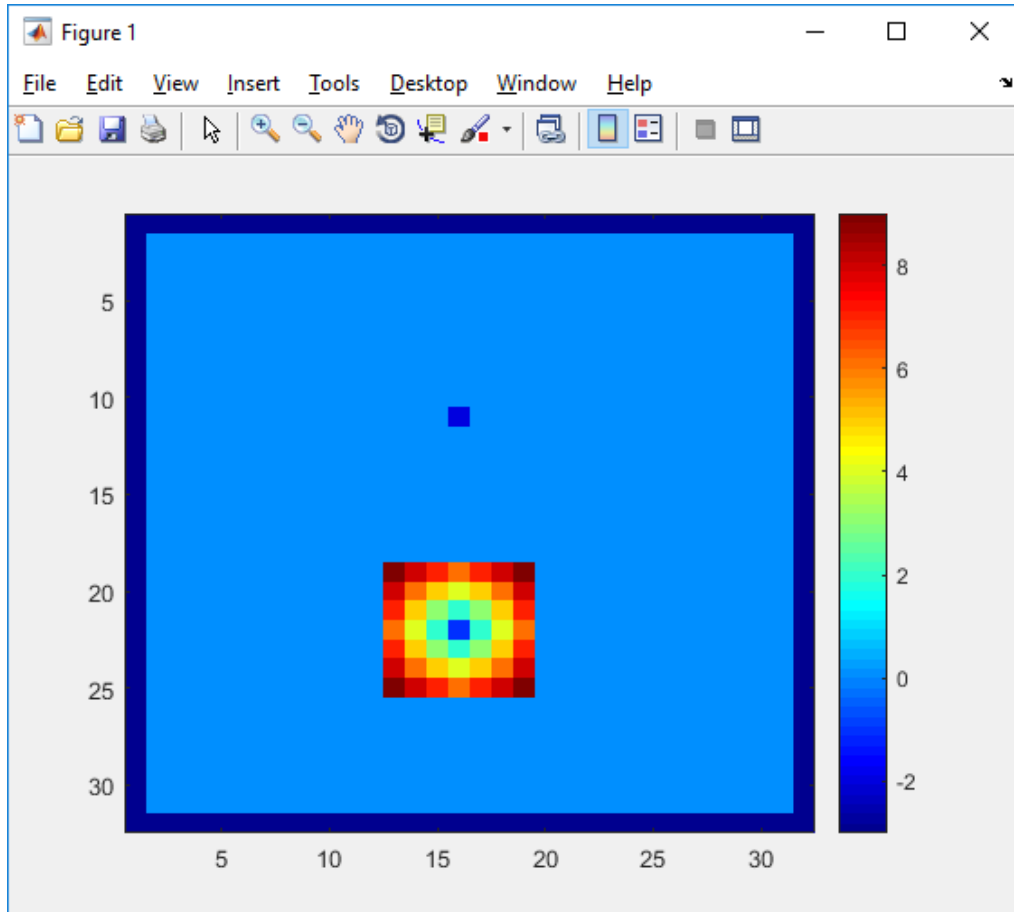
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	3	2	3	0	0	0
0	0	2	-1	2	0	0	0
0	0	3	2	3	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	6	5	4	5	6	0	0
0	5	3	2	3	5	0	0
0	4	2	-1	2	4	0	0
0	5	3	2	3	5	0	0
0	6	5	4	5	6	0	0
0	0	0	0	0	0	0	0

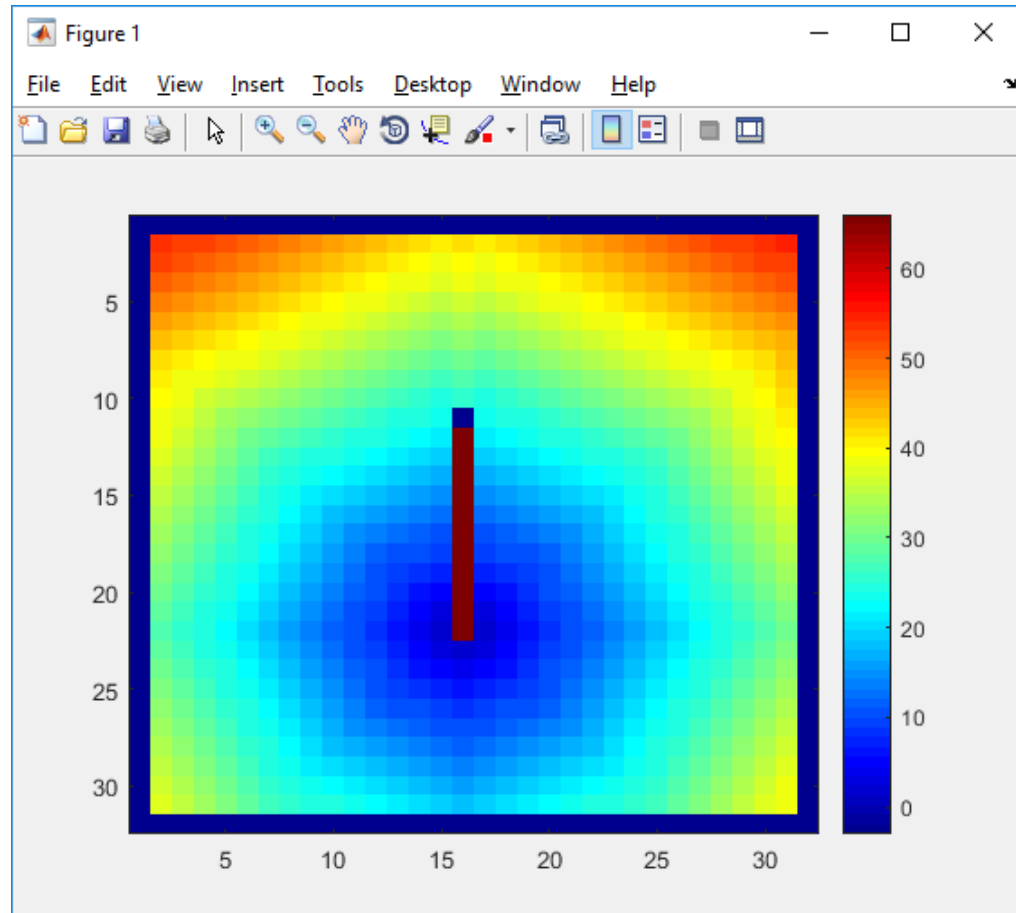
9	8	7	6	7	8	9
8	6	5	4	5	6	8
7	5	3	2	3	5	7
6	4	2	-1	2	4	6
7	5	3	2	3	5	7
8	6	5	4	5	6	8
9	8	7	6	7	8	9







Végére...



- Ha a hullámfront eléri a robotot: Van út
- A robot a környezetében (8 szomszédos cella) megkeresi a legkisebb hullámfront cellát, és oda lép
- Véges sok lépés után a legrövidebb úton (mert az átlós irányt is figyelembe vettük, és ismert terepen vagyunk) eléri a célt

