

Mesterséges Intelligencia Alapok

Első Beadandó feladat

Levelező Tagozat

2024-25/I. félév

Az első beadandó feladat célja az egyszerű gráfkereső algoritmusok megismerése – a köztük lévő kis, kódbeli változtatások hogyan befolyásolják a működésüket.

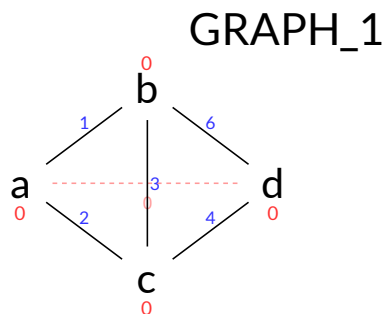
A feladatot nagyban leegyszerűsíti, hogy Java JUnit tesztek is készültek az egyes algoritmusokhoz. A mélységében először (DFS) és széltében először (BFS) algoritmusok már implementálva lettek. A feladat a nyalábkeresés (Beam search), hegymászó algoritmus (Hill-climbing), legjobbat először keresés (Best-first search), elégazás és lenyesés (Branch and Bound), annak heurisztikával (Branch and Bound with Heuristic) és kiterjesztési listával (Branch and Bound with extended set/list), valamint az A* algoritmusok implementációja Java nyelven, a meglévő „keretrendszer” használva. Az előforduló egyenlőségeket lexikográfiai sorrendben oldjuk fel.

A feladat csak akkor elfogadható, ha a minden JUnit teszt megfelelően, hiba nélkül lefut a három példagráfon.

A gráfok

A feladat három gráfot tartalmaz, melyekre egységesen is készült. Mindhárom gráf egyedi tulajdonságokkal rendelkezik, hogy kimerítőbb teszteket tudjunk velük elvégezni.

Az első gráf egy egyszerű példa arra, hogy az egyes algoritmusok hogyan viselkednek, amikor a heurisztikát kivesszük (egységesen nullára állítjuk). a-ból akarunk eljutni d-be. Figyeljük meg, hogy mely algoritmusok adják vissza ugyanazokat az útvonalakat és miért.



1. ábra: Az első (egyszerű) gráf

A második gráf egy nagyobb példa konzisztens heurisztikával. Itt az algoritmusok már javarészt eltérnek egymástól. S-ből akarunk

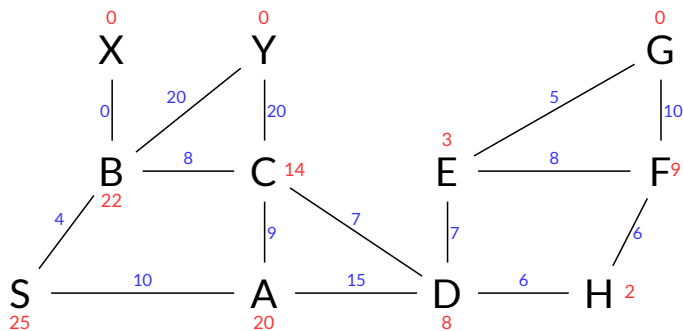
Algoritmus	Talált útvonal
Mélységében először	abcd
Széltében először	abd
Nyaláb ($w = 2$)	abd
Hegymászó	abcd
Legjobbat először	abcd
Elágazás és lenyesés	acd
Elágazás és lenyesés listával	acd
Elágazás és lenyesés heurisztikával	acd
A*	acd

1. táblázat: Az első gráf (1. ábra) útvonalai

eljutni G-be. Figyeljük meg, hogy melyik algoritmussal hány iteráció alatt tudtuk megtalálni a G-t.

GRAPH_2

2. ábra: A második példa gráf



A harmadik gráf a heurisztikát teszteli, hiszen itt nem konzisztensen lett megválasztva. Ennek eredménye, hogy az informált kereső algoritmusok nem feltétlen találják meg a legrövidebb utat.

Elfogadhatóság:

$$H(X,G) \leq D(X,G),$$

azaz X csúcs heurisztika szerinti távolságának kisebbnek vagy egyenlőnek kell lennie a céltól (G -től) való tényleges legrövidebb távolságával. Röviden: A heurisztikának alábecslésnek kell lennie.

Konzisztencia:

$$|H(X) - H(Y)| \leq D(X,Y),$$

azaz két pont heurisztikájának a távolsága kisebb vagy egyenlő kell legyen a köztük lévő legrövidebb útvonal hosszánál. Például a 3. gráfban (3. ábra):

Algoritmus	Talált útvonal
Mélységében először	SACDEFG
Széltében először	SACEG
Nyaláb ($w = 2$)	SBYCEG
Hegymászó	SADHFG
Legjobbát először	SADEG
Elágazás és lenyesés	SBCEG
Elágazás és lenyesés listával	SBCEG
Elágazás és lenyesés heurisztikával	SBCEG
A*	SBCEG

2. táblázat: Az második gráf (2. ábra) útvonalai

$$|H(z) - H(x)| \leq D(z, x)$$

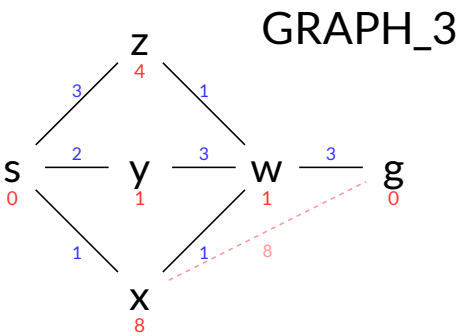
$$4 \not\leq 2$$

Ez mutatja, hogy a heurisztika nem konzisztens.

$$H(x, g) \leq D(x, g)$$

$$8 \not\leq 4$$

pedig mutatja, hogy a választott heurisztika nem elfogadható. A nem elfogadhatóságból egyébként következik, hogy nem konzisztens a heurisztika. Az A* kereső esetén feltétel a konzisztencia, az elágazás és lenyesésnél pedig ha heurisztikát alkalmazunk, akkor az elfogadhatósági feltételt kell teljesíteni.



3. ábra: A harmadik gráf, amely inkonzisztens heurisztikát tartalmaz

Az implementációhoz szabadon használható online segédlet, például a tantárgy gyakorlati oldala, a tantárgy dokuwiki oldala. ChatGPT használata csak pontos megjelöléssel engedett: prompt, válasz leírása.

Algoritmus	Talált útvonal
Mélységében először	sxwg
Széltében először	sxwg
Nyaláb ($w = 2$)	sywg
Hegymászó	sywg
Legjobbat először	sywg
Elágazás és lenyesés	sxwg
Elágazás és lenyesés listával	sxwg
Elágazás és lenyesés heurisztikával	szwg
A*	sywg

3. táblázat: A harmadik gráf (3. ábra) útvonalai