**Mélységi és szélességi keresés**

Kereső fával kereső algoritmusok

Közös jellemző, hogy a technikai művelet a kiterjesztés. Az adatbázis a gráf eddig feltárt részének a feszítőfája. Ha belegondolunk itt a legnagyobb az adatbázis. Sokkal több memóriára van szükség. A kiterjesztés azt jelenti, hogy egy csúcsba alkalmazom az összes alkalmazható operátort, azaz feltárom a gyermek csúcsait.

* Mélységi keresés – Depth-first search
* Szélességi keresés – Breadth-first search
* Optimális keresés – Optimal-first
* Best-first keresés
* A-algoritmus
* A\*-algoritmus

## Kereső-fával kereső algoritmusok közös váza

**Adatbázis**: A gráf eddig feltárt része feszítő faként tárolva. Ezt a feszítőfát két halmazban tárolom el. A nyílt (NY) és a zárt (Z) csúcsok halmazaként. A zárt csúcsok azok a csúcsok, amiket már felderítettem és kiterjesztettem. A nyílt csúcsok a már felderített, de nem kiterjesztett csúcsok. A keresést mindig valamelyik nyílt csúcsból folytatom.

**Inicializálás**: Zárt csúcsok halmaza legyen egyenlő üres halmazzal. Z = { }. S = (k, kif(S), NULL), ahol k = állapot, kif() = kiértékelő függvény más és más lesz minden függvényben, NULL = szülő. NY = { S}, ahol S a start csúcs.

**Ciklus**:

Első lépés: a nyílt csúcsok közül kiválasztom azt, amelynek a kiértékelő függvény értéke a legkisebb, vagy a legnagyobb. Ezt a csúcsot úgy hívom, hogy kiterjesztésre kiválasztott csúcs.

Második: A kiterjésztésre kiválasztott csúcsot kiterjesztem, így új csúcsokat kapunk.

Harmadik lépés: Az új csúcsokat felveszem a nyílt csúcsok közé, abban az esetben, ha nem szerepeltek még sem a nyíltak, sem a zártak közt, úgy értve, hogy az új csúcs állapota nem található meg sem a nyílt csúcsok álalotai között, sema zárt csúcsok állapota között.Tehát nem két csúcsot hasonlítok össze, hanem a két csúcs álalpotát hasonlítom össze.

Negyedik lépés: Ha valamely új csúcs már szerepel a nyíltak között, akkor ezt az új csúcsot vagy eldobom, vagy csere.

Ötödik lépés: ha egy új csúcs szerepel a zártak között akkor az új csúcsotvagy eldobom, vagy csere.

Hatodik: a kiterjesztett csúcsot átminősítem zárttá, azaz kiveszema nyíltak közül és beteszem a zártak közé.

**Pozitív kilépési feltétel**: Lehet gyors vagy lassú.

Gyors, pozitív kilépési feltétel: Ha az új csúcsok között van terminális, akkor megvan a megoldás. A megoldás adódik terminális csúcs szülő referenciáján keresztül visszafelé haladva.

Lassú pozitív kilépési feltétel: Ha a kiterjesztésre kiválasztott csúcsterminális, akkor megoldás adódik terminális csúcs szülő refernciáin visszafelé haladva.

A pozitív kilépési feltétel vagy lassú vagy gyors.

**Negatív kilépési feltétel**: ha a nyílt csúcsok halmaza üres, akkor azt írom ki, hogy nincs megoldás.

**Technikai művelet:** Kiterjesztés. A kiterjesztésre kiválasztott csúcsra alkalmazom az összes alkalmazható operátort. Így új állapotokat kapok. Az új állapotból új csúcsot hozok létre úgy hogy:

Új csúcs = (új állapot, kif(újcsúcs), kiterjesztésre kiválasztott csúcs)

3 kérdés:

1. Igen, mert szisztematikus és van benne körfigyelés. Van benne körfigyelés, hiszen a ciklus harmadik pontjában, csak azokat az új csúcsokat veszem fel a nyíltak közé amik idáig nem szerepeltek sem a nyíltak sem a zártak között.
2. IGEN, mert szisztematikus és van benne kifigyelés.
3. Van olyan változat, ami optimális, van olyan, ami nincs.

Erős algoritmus, mára közös váz is igen-igen igen algoritmus.

Az egyes algoritmusok a pirossal jelölt helyeken különbözhetnek. A legmarkánsabb különbség az egyes algoritmusok között az h mi a kiértékelő függvény. A ciklus első lépésébe vagy min-t választok vagy maxot. Csak 1 helyen választok maxot.

Táblázat:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Magyarázat |  | Legkissebb/legnagyobb mélységben lévő csúcsot választja ki | új eleme nyílt | új eleme zárt | Lehet a gyorsat használni, de nem muszáj. Ahol lassú van, ott csak a lassút lehet. |  |  |
| Kereső fával keresők | Kiértékelő függvény | Min/Max | Új E NY | Új E Z | Pozitív kilpési feltétel | optimális | Milyen értelemben? |
| Mélységi | m(A) | MAX \* | Eldobom | Eldobom | Gyors | nem |  |
| Szélességi | m(A) | MIN | Eldobom | Eldobom | Gyors | igen | Legkisebb mélységű megoldást találja |
| Optimális | g(A) | MIN | Ha új jobb, akkor csere | Eldobom | Lassú | igen | Legkisebb út költségű megoldást találja |
| Best-First | h(A) | MIN | Eldobom | Eldobom | Gyors | nem |  |
| A-algoritmus | f(A) | MIN | Ha új jobb, akkor csere | Ha új jobb, akkor csere és vissza minősítés | Lassú | nem |  |
| A\*-algoritmus | f(A) h(A) <= h\*(A) | MIN | Ha új jobb, akkor csere | Ha új jobb, akkor csere és vissza minősítés | Lassú | igen | A legkisebb út költségű megoldás |

Mélység definiálása:

M(S) = 0

m(A) = m(A.szülő) +1

\*Olvasva: A mélységi keresés mindiga legnagyobb mélységűnyílt csúcsot választja kiterjesztésre.

DEF(Út költség)

g(A): (: azt jelenti h jelentése): S-ből A-ba vezető út költsége.

g\*(A): Start csúcsból az A csúcsba vezető **optimális** **út költsége.**

g(S)= 0 (start csúcsból strat csúcsba vezeteő)

g(A) = g(A.szülő) + r(o), ahol az A.szülőből az o operátor segítségével jutottam A-ba, és r(o) e ennek a költsége.

MEGJ: r(o) > 0 „There is no free lunch”, azaz egyik operátor költsége sem lehet 0, vagy negatív.

1. ábra

g(A) >= g\*(A)

Def (Heurisztika)

h(A): Aktuálisból, A-ból legközelebbi terminális csúcsba vezető út becsült költsége.

h\*(A): Aktuálisból, A-ból legközelebbi terminális csúcsba vezető **optimális út költsége**.

1. ábra

Definiálni nem tudom, csak a tulajdonságait tudom elmondani.

h(T) ~ 0 Terminálisból terminálisba vezető költség 0. ~ hullámos egyenlőség jel lenne

h(A) ~ h\*(A)

DEF

f(A) = g(A) + h(A)

Múlt + jövő. 2. ábra.

f(A): S-ből A-n keresztül a legközelebbiterminális csúcsba vezető út becsült költsége.

f\*(A): S-ből A-n keresztül legközelebbi terminális csúcsba vezető optimális út költsége.

f\*(S) = g\*(S) +h\*(S) = **h\*(S)** (a g\*(S) = 0) **= Az optimális megoldás költsége.**

Mikor egyenlő f\*(A) az f\*(A)-el?

Mikor f\*(A) = f\*(S) ? Ha az A csúcs rajta van az optimális megoldáson







