

Generikus irányítatlan gráf tervek

Generikus irányítatlan gráf tervek

Osztályok

Felhasznált osztályok

Graph Osztály

Edge Osztály

Vertex Osztály

VertexSet Osztály

BFFSet Osztály

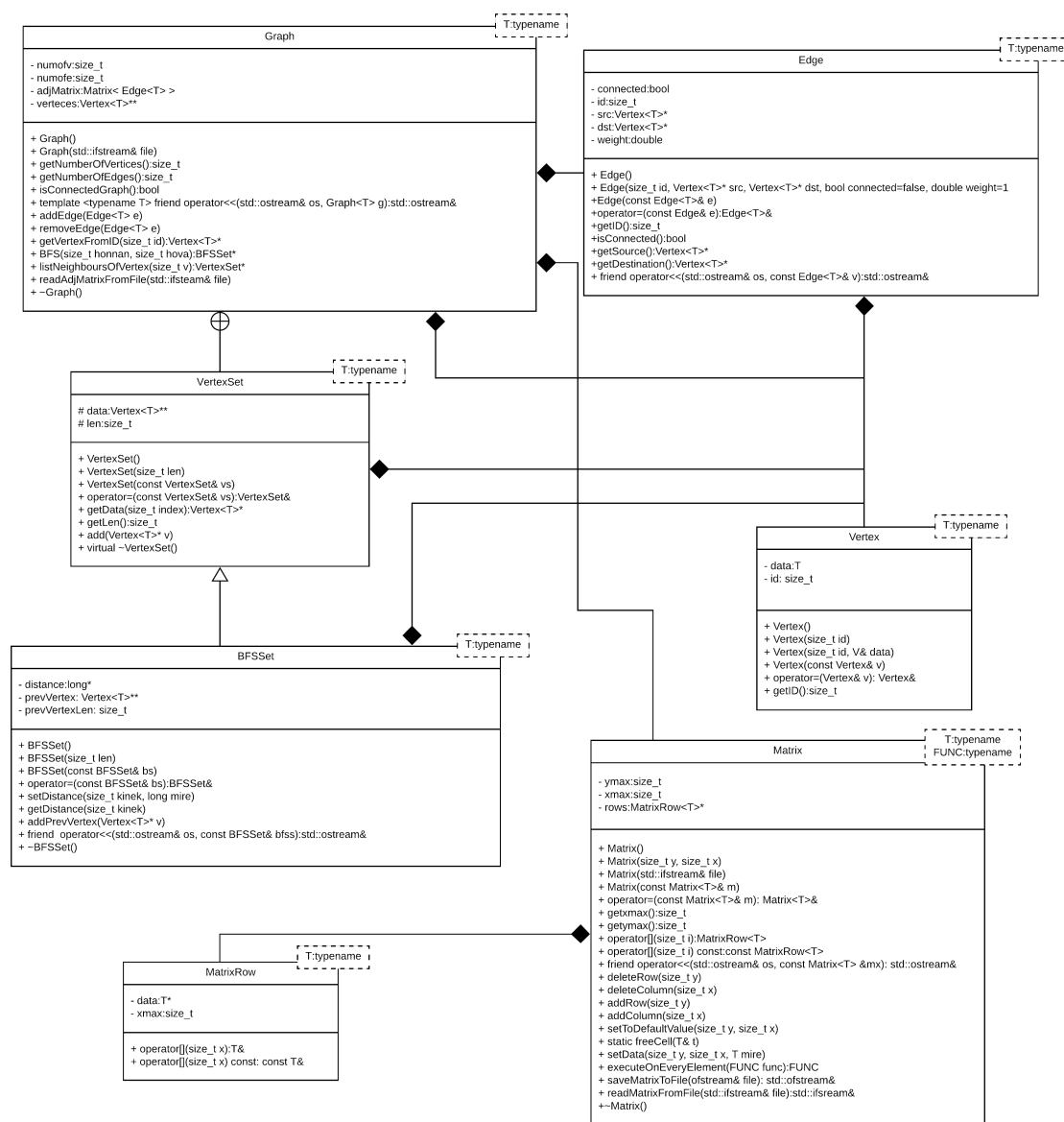
Matrix Osztály

MatrixRow Osztály

Bővíthetőség

Felhasznált algoritmus

Osztályok



A feladat megvalósításához előreláthatólag 7 db osztályt használok fel. A tanult eszközök közül az öröklést, az egymásba ágyazott osztályokat és a tartalmazást tervezem felhasználni. Az ábrán még nem szerepel, de a diagram bővülni fog a teszteléshez használt példa osztályokkal.

Felhasznált osztályok

Graph Osztály

Ez az osztály fogja össze a maradék 6 db osztályt. Ez az osztály tartalmazza, a feladatkiírásban szereplő feladatokat megvalósító függvényeket. A legtöbb feladat több segédfüggvényre van bontva.

Edge Osztály

Ez az osztály foglalja össze az élekhez tartozó információkat. Az osztály megkülönbözteti az élnek a forrás és cél csúcsát. Ezzel a felkészítve arra, hogy a jövőben irányított gráfokat is tudjon kezelni. A `weight` változó lehetővé teszi, hogy az éleket súlyokkal lássuk el, ez szintén a jövőbeli továbbfejlesztés lehetőségét szolgálja.

Vertex Osztály

Ez az osztály tárolja a csúcsokhoz tartozó adatokat. Mindent csúcsot ellátok egy azonosítóval. Ez a BFS algoritmus során segít a csúcsok megkülönböztetésében és így tudok valamilyen módon hivatkozni a csúcsokra.

VertexSet Osztály

Ezt az osztályt azért hoztam létre, hogy a függvényekből visszatérő adatokat egységbe tudjam foglalni. Így a dinamikus tömb és annak mérete egymás mellett tárolódik, könnyen elérhető. Az osztály a `Graph` osztály részét képezi, mert logikailag oda tartozik.

BFFSet Osztály

Ez az osztály a `VertexSet` osztályból származik. A célja hasonló. Ez kifejezetten a BFS algoritmus visszatérési értékének a tárolója.

Matrix Osztály

Ez az általános felhasználásra szánt osztályt felelős a fájlból történő beolvasás mátrixának tárolására. Az osztály tervezése során felhasználtam a programozás 1 tárgy keretein belül készített házim(mátrix függvénykönyvtár) létrehozása során szerzett tapasztalataimat.

MatrixRow Osztály

Ez az osztály azért jött létre, hogy a mátrix osztály kettős indexelését lehetővé tegye. Ezen kívül ennek az osztálynak a segítségével elkerülhető, hogy a Mátrix osztályban `T***` típusú adatot kelljen tárolni.

Bővíthetőség

A program tervezése során lehetőséget biztosítottam a jövőbeli bővíthetőségre és továbbfejlesztésre.

- A mátrix és a gráf generikus, így bármilyen adatot képes tárolni
- Az élek fel vannak készítve irányított gráfok által való használatra
- Az élekben található `weight` paraméter lehetőséget teremt, hogy a gráf éleit súlyokkal lássuk el.

Felhasznált algoritmus

A specifikációban már említettem, hogy a program célja a [BSZ2](#) tárgyból tanultak elmélyítése. Így a szélességi bejárás algoritmusát is a tárgyban tanultak alapján szeretném megvalósítani. Az algoritmus és annak teljes leírása [itt](#) érhető el. Szeretném kiemelni az alábbi fontosabb részt:

- $b(i)$ ($i = 1, 2, \dots$): az i -ediként bejárt csúcs
- $t(v)$ ($v \in V$): v távolsága s -től
- $m(v)$ ($v \in V, v \neq s$): a v -t megelőző csúcs az algoritmus által megtalált, s -ből v -be vezető legrövidebb úton
- j : az eddig bejárt csúcsok száma
- k : a jelenleg aktív csúcs sorszáma a $b(1), b(2), \dots$ sorozatban

BFS ALGORITMUS

Bemenet: Egy $G = (V, E)$ gráf és egy $s \in V$ csúcs

```
1   $j \leftarrow 1; k \leftarrow 1; b(1) \leftarrow s$ 
2   $t(s) \leftarrow 0$ ; minden  $v \in V, v \neq s$ -re  $t(v) \leftarrow *$ 
3  ciklus
4      ha a  $b(k)$  csúcsnak van olyan  $v$  szomszédja, amelyre  $t(v) = *$ , akkor:
5           $j \leftarrow j + 1$ 
6           $b(j) \leftarrow v$ 
7           $t(v) \leftarrow t(b(k)) + 1$ 
8           $m(v) \leftarrow b(k)$ 
9      különben:
10         ha  $k = j$ , akkor:
11             stop
12         különben:
13              $k \leftarrow k + 1$ 
14  ciklus vége
```

Az algoritmusom megvalósítása közben hasonló változóneveket szeretnék használni. Az algoritmust jelenleg egy helyen tervezem módosítani. A $*$ távolság helyett -1 -et tervezek használni. Nálam ez lesz a végtelen jelölése, ez hibakeresés során jól megkülönböztethető lesz a többi távolság értéktől.