

Gráfalgoritmusok és hatékony adatszerkezetek szemléltetése

Készítette: Bognár Gergő
Témavezető: Veszprémi Anna

Eötvös Loránd Tudományegyetem Informatikai Kar
Algoritmusok és Alkalmazásai Tanszék

Budapest, 2012.

A dolgozat témája

Gráfalgoritmusok

- Páros gráfok és maximális párosítás
- Minimális költségű feszítőfa illetve feszítő erdő keresés

Grafikus felület

- Gráf megjelenítés, szerkesztés
- Algoritmus szemléltetés
- Adatszerkezetek bemutatása

Célkitűzés

- Szemléltetés, bemutatás
- Tanulási segédanyag

A dolgozat témája

Gráfalgoritmusok

- Páros gráfok és maximális párosítás
- Minimális költségű feszítőfa illetve feszítő erdő keresés

Grafikus felület

- Gráf megjelenítés, szerkesztés
- Algoritmus szemléltetés
- Adatszerkezetek bemutatása

Célkitűzés

- Szemléltetés, bemutatás
- Tanulási segédanyag

A dolgozat témája

Gráfalgoritmusok

- Páros gráfok és maximális párosítás
- Minimális költségű feszítőfa illetve feszítő erdő keresés

Grafikus felület

- Gráf megjelenítés, szerkesztés
- Algoritmus szemléltetés
- Adatszerkezetek bemutatása

Célkitűzés

- Szemléltetés, bemutatás
- Tanulási segédeszköz

Páros gráfok témakör

Párosság vizsgálata

- Csúcsok felosztása, pirosra és kékre színezése
- Alapja: szélességi bejárás
- Sor adatszerkezet

Magyar módszer

- Maximális párosítás keresése páros gráfban
- Ötlet: párosítás növelése javító úttal
- Alternáló erdő építés
- Szélességi bejárás speciális alkalmazása
- Sor adatszerkezet

Páros gráfok témakör

Párosság vizsgálata

- Csúcsok felosztása, pirosra és kékre színeezése
- Alapja: szélességi bejárás
- Sor adatszerkezet

Magyar módszer

- Maximális párosítás keresése páros gráfban
- Ötlet: párosítás növelése javító úttal
- Alternáló erdő építés
- Szélességi bejárás speciális alkalmazása
- Sor adatszerkezet

Minimális költségű feszítőfa témakör

Piros-kék eljárás

- Piros és kék szabály

Kruskal algoritmus

- Mohó színezési stratégia, legkisebb súlyú szintelen él színezése
- Kék fák, diszjunkt halmaz műveletek
- Unió-Holvan adatszerkezet

Prim algoritmus

- Mohó stratégia, minden lépésben a kék szabály alkalmazása
- Csúcsok aktuális erdőtől vett távolsága
- Kupac adatszerkezet

Minimális költségű feszítőfa témakör

Piros-kék eljárás

- Piros és kék szabály

Kruskal algoritmus

- Mohó színezési stratégia, legkisebb súlyú szintelen él színezése
- Kék fák, diszjunkt halmaz műveletek
- Unió-Holvan adatszerkezet

Prim algoritmus

- Mohó stratégia, minden lépésben a kék szabály alkalmazása
- Csúcsok aktuális erdőtől vett távolsága
- Kupac adatszerkezet

Minimális költségű feszítőfa témakör

Piros-kék eljárás

- Piros és kék szabály

Kruskal algoritmus

- Mohó színezési stratégia, legkisebb súlyú szintelen él színezése
- Kék fák, diszjunkt halmaz műveletek
- Unió-Holvan adatszerkezet

Prim algoritmus

- Mohó stratégia, minden lépésben a kék szabály alkalmazása
- Csúcsok aktuális erdőtől vett távolsága
- Kupac adatszerkezet

Gráfok tulajdonságai

Vizsgált gráfok

- Irányítatlan, egyszerű, élsúlyozott vagy élsúly nélküli gráfok
- Csúcsok címkézése egyedi sorszámmal
- Élek súlyozása egész számmal

Algoritmusok speciális viselkedése

- Nem összefüggő gráfokra is működő algoritmusváltozatok
 - Szélességi bejárás
 - Prim algoritmus

Gráfok tulajdonságai

Vizsgált gráfok

- Irányítatlan, egyszerű, élsúlyozott vagy élsúly nélküli gráfok
- Csúcsok címkézése egyedi sorszámmal
- Élek súlyozása egész számmal

Algoritmusok speciális viselkedése

- Nem összefüggő gráfokra is működő algoritmusváltozatok
 - Szélességi bejárás
 - Prim algoritmus

Program és kód

Környezet, futtatás

- Java virtuális gép
- JAR archívum (*graf.jar*)
- Indítás virtuális géppel, vagy appletként

Fejlesztés

- Objektorientált, eseményvezérelt
- Grafikus felület
- Logikailag egy végrehajtási szál
- Csomagok, osztályok, erőforrásfájlok

Program és kód

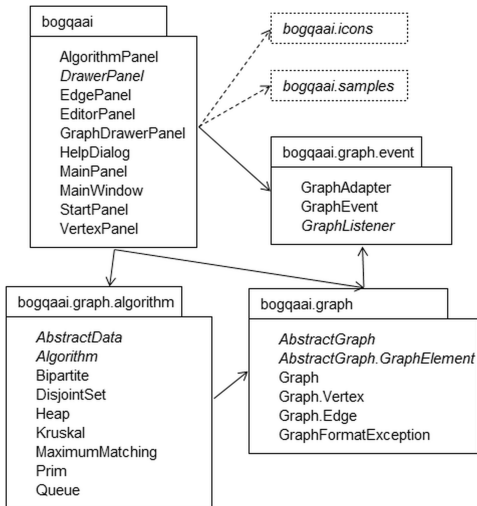
Környezet, futtatás

- Java virtuális gép
- JAR archívum (*graf.jar*)
- Indítás virtuális géppel, vagy appletként

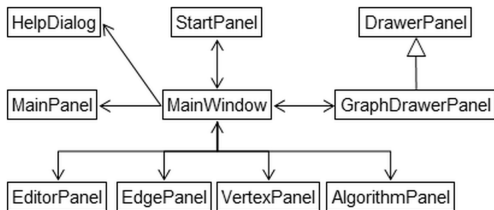
Fejlesztés

- Objektorientált, eseményvezérelt
- Grafikus felület
- Logikailag egy végrehajtási szál
- Csomagok, osztályok, erőforrásfájlok

Csomagok

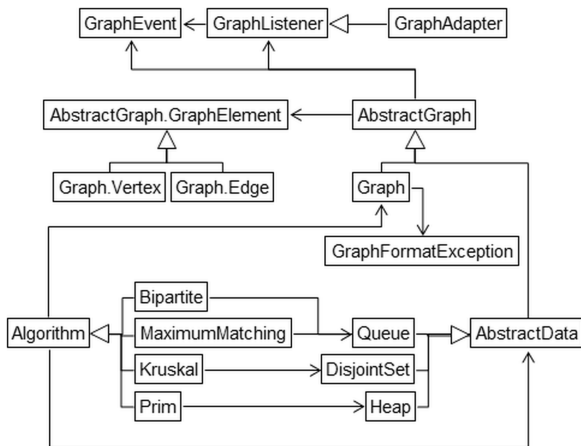


Grafikus felület



A grafikus felület osztályai

Belső ábrázolás



Gráfok, algoritmusok és adatszerkezetek osztályai

Algoritmus osztály

Algorithm

```
+ int UNKNOWN
+ int FAIL
+ int SUCCESS
# java.util.ResourceBundle bundle
- String initFormat
- String stepFormat
- String resultFormat
# Graph graph
# AbstractData data
# int result
- int step
- String text
```

```
+ Algorithm(Graph graph)
+ boolean isGraphWeighted()
# String internalReset()
# String internalStep()
+ void reset()
+ void step()
+ void run()
+ void changed()
+ void animation(float percent)
+ AbstractData getData()
+ boolean isReady()
+ int getResult()
+ String getText()
+ <T extends Algorithm> Strings
  getStrings(Class<T> clazz)
```

Példa: Kruskal osztály

```
/** Az algoritmus léptetése. */  
@Override  
protected String internalStep() {  
    String resultText;  
    if(edge < edges.size()) {  
        Edge e = edges.get(edge);  
        Graph.Vertex u = e.edge.getVertexU();  
        Graph.Vertex v = e.edge.getVertexV();  
        resultText = java.text.MessageFormat.format(stepMessage, u.toString(), v.toString());  
        if(set.find(u) != set.find(v)) {  
            e.edge.set(Graph.Color.BLUE, 0, Graph.Animation.NORMAL);  
            set.union(u, v);  
        } else {  
            e.edge.set(Graph.Color.RED, 0, Graph.Animation.NORMAL);  
        }  
        ++edge;  
    } else {  
        result = SUCCESS;  
        resultText = resultMessage;  
        for(Edge e: edges) {  
            if(e.edge.getColor() == Graph.Color.RED) {  
                e.edge.set(Graph.Color.GRAY, 0);  
            }  
        }  
    }  
    return resultText;  
}
```

Fejlesztési lehetőségek

Bővíthetőség

- Mintagráf hozzáadása
- Algoritmus hozzáadása
- Adatszerkezet hozzáadása

Dokumentáltság

- Javadoc

Köszönöm a figyelmet!