



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

Modelando problemas problemas con grafos

21 de octubre de 2018

Algoritmos y Estructura de Datos III

Integrante	LU	Correo electrónico
Buceta, Diego	001/17	diegobuceta35@gmail.com
Springhart, Gonzalo	318/17	glspringhart@gmail.com

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

1. Introducción al problema

2. Justificación teórica

3. Algoritmos presentados

3.1. Kruskal

KRUSKALSINPATHCOMP(*ListaIncidencia* : *grafoCompleto*, *cantNodos* : *entero*)

```
1 padre ← vector de enteros de tamaño de cantNodos y cargado con el valor de su posición en cada posición
2 AGM ← lista de incidencia vacía, de tamaño cantNodos-1
3 OrdenarPorPeso(grafoCompleto)
4 for e:Arista ∈ grafoCompleto
5   if getPadre(indice(e.primerNodo), padre)=getPadre(indice(e.segundoNodo), padre)
6     agregar(e, agm)
7
8
9 Devolver AGM
```

GETPADRE(*entero* : *indice*, *padre* : *vectordeenteros*)

```
1 if padre[indice] == indice
2   Devolver indice
3 else
4   getPadre(indice(padre[indice]), padre)
5
```

GETPADRECONPATHCOMP(*entero* : *indice*, *padre* : *vectordeenteros*, *altura* : *vectordeenteros*, *nivelesSubidos* : *entero*)

```
1 if padre[indice] == indice
2   altura[indice] = nivelesSubidos
3   Devolver indice
4 else
5   padre[indice] = getPadreConPathComp(indice(padre[indice]), padre, altura, nivelesSubidos+1)
6   Devolver padre[indice]
7
```

UNIRPADRES(*indiceNodo1* : *entero*, *indiceNodo2* : *entero*, *padre* : *vectordeenteros*)

```
1 padreNodo1 ← getPadre(indiceNodo1, padre)
2 padre[indiceNodo2]=padreNodo1
```

UNIRPADRESCONPATHCOMP(*indiceNodo1* : *entero*, *indiceNodo2* : *entero*, *padre* : *vectordeenteros*, *altura* : *vectordeenteros*, *nivelesSubidos* : *entero*)

```
1 padreNodo1 ← getPadreConPathComp(indiceNodo1, padre, altura, 0)
2 padreNodo2 ← getPadreConPathComp(indiceNodo2, padre, altura, 0)
3 padreMenosAltura ← min(altura[padreNodo1], altura[padreNodo2])
4 padreMasAltura ← max(altura[padreNodo1], altura[padreNodo2])
5 padre[padreMenosAltura]=padreMasAltura
```

ARMARGRAFOCOMPLETO(*nodos : vectordeNodos*)

```

1  listaAristas ← inicializar lista de incidencia
2  matrizAristas ← inicializar matriz de adyacencia
3  for  $i \leftarrow 0$  to tam(nodos)
4  for  $j \leftarrow i + 1$  to tam(nodos)
5  armar arista con datos de v[i] y v[j]
6  agregar arista a listaAristas
7  armar matriz de adyacencia con la lista de incidencia
8  Devolver Matriz de adyacencia y Lista de incidencia

```

RETIRAREJESINCONSISTENTES(*listaAristas : lista incidencia*, σ_T , *profVecindario*, f_T , *forma*, *cantidadDeClusters*, *vectordeenteros*)

```

1  for  $e : listaAristas$ 
2  calcular media y desviacion respecto del vecindario de profVecindario de profundidad de cada extremo de e. (u
3  si es inconsistente
4  sacar e de las listas
5  recorrer en la lista de ady todos los nodos alcanzables de uno de los extremos y modificar su representante en
6  aumentar en 1 el valor de cantidadDeClusters
7

```

3.2. Complejidad

4. Experimentación

4.1. Variaciones

Los experimentos estarán centrados en analizar las diferentes tipos clusterizaciones que pueden realizarse variando las definiciones de eje inconsistente. Intentaremos analizar las configuraciones necesarias para que se pueda alcanzar una clusterización lo más cercana a la de la percepción humana y los resultados interesantes al que pueden llegarse.

Dados los siguientes: f_T multiplicador del promedio, σ_T multiplicador de la desviación, y la profundidad del vecindario de los extremos del eje candidato, $W(XY)$ el peso del eje candidato, y sea X e Y sus nodos extremos, definiremos un eje inconsistente:

- Forma 1: $\frac{W(XY)}{\text{Promedio}(\text{Vecindario}(X))} > f_T$ y $\frac{W(XY)}{\text{Promedio}(\text{Vecindario}(Y))} > f_T$,
Es decir, la proporción entre el peso del eje candidato y el promedio de peso del vecindario de sus extremos es mayor al coeficiente dado.
- Forma 2: $W(XY) > \text{Promedio}(\text{Vecindario}(X)) + \sigma_T * \text{desviacion}(\text{Vecindario}(X))$
y $W(XY) > \text{Promedio}(\text{Vecindario}(Y)) + \sigma_T * \text{desviacion}(\text{Vecindario}(Y))$,
Es decir, que el peso del eje candidato supere al promedio del vecindario de sus extremos por al menos σ_T unidades de la desviación del vecindario del extremo.
- Forma 3: Que se cumpla ambas