



Componentes Conexas

Laboratorio Semana 5

1 Introducción

La Internet Profunda, o *Deep Web*, se compone de todas las páginas web no alcanzables por los buscadores. Si los nodos de su grafo son páginas web, y las aristas son hipervínculos, puede encontrar cuáles páginas web pertenecen a la internet profunda buscando la componente conexas, usando los algoritmos de búsqueda en profundidad (*Depth-First Search*) y búsqueda en amplitud (*Breadth-First Search*).

Debido a que la red de páginas web puede ser muy compleja, es necesario diseñar el algoritmo para que pueda ser truncado: debe poder evitar que el algoritmo vaya a más de una profundidad dada desde el vértice inicial. También es útil poder ver cómo se recorrieron los vértices: esto puede hacerse imprimiendo el árbol, donde el nivel de indentación (número de tabuladores \t) indica el nivel de profundidad; imprimiendo el ordinal de visita de cada vértice, indicado para cada vértice, el orden en que fue visitado (primero, segundo, etc); o imprimiendo los predecesores de cada vértice, mostrando a través de cuál vértice fue alcanzado.

2 Descripción de la Actividad

Se desea que cree una aplicación que recorra un grafo a partir de un nodo (resultado del buscador) e indique cuáles nodos no son parte de la componente conexas de ese nodo (son de la Internet Profunda). El programa deberá poder manejar opciones de consola que trunquen el recorrido, impriman el árbol de recorrido, el ordinal de visita de cada vértice y/o los predecesores del vértice.

Debe crear un archivo llamado `DeepWeb.java` con la implementación de la aplicación.

3 Entrada de datos

El programa se debe poder ejecutar desde la consola con el siguiente comando:

```
>java DeepWeb <instancia> <origen> <dfs/bfs> [--trunc #] [--arb] [--ord] [--pred]
```

donde

- `<instancia>` es el nombre del archivo que contiene el grafo,
- `<origen>` es el número entero que corresponde a la página web retornada por el buscador,
- `<dfs/bfs>` es una de las palabras clave indicadas: “dfs” ó “bfs”, indicando cuál de los algoritmos se usará
- `--trunc #` (la palabra clave “--trunc” seguida de un número entero) indica a qué profundidad se debe detener el recorrido,
- `--arb` (la palabra clave “--arb”) indica que debe imprimirse el árbol del recorrido,
- `--ord` (la palabra clave “--ord”) indica que debe imprimirse el ordinal de cada vértice, y
- `--pred` (la palabra clave “--pred”) indica que debe imprimirse el predecesor de cada vértice.

Recuerde que puede usar `String.equalsIgnoreCase()` y `Integer.parseInt()`.

El formato del archivo que contiene los datos de un grafo es el siguiente:

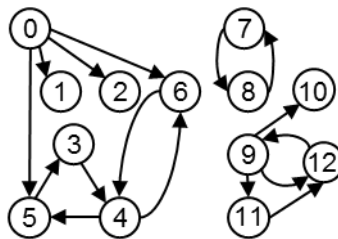
```
NumeroDeVertices
NumeroDeLados
idVerticeInicial1 idVerticeFinal1
:
idVerticeInicialNumeroDeLados idVerticeFinalNumeroDeLados
```

4 Salida de los Datos

Si existen nodos no-alcanzables, se deben imprimir, en orden, separados por comas. En caso de que todos los vértices hayan sido alcanzados, se debe imprimir “Todas las paginas son parte de la internet visible.”

5 Ejemplos

Considere el siguiente grafo dirigido, el cual se proporciona en el archivo `ejemplo1.txt`:



El resultado de la llamada `DeepWeb ejemplo1.txt 0 dfs --ord --pred` debería ser:

```
7,8,9,10,11,12
```

Ordinales:

```
0: 0
1: 1
2: 2
3: 4
4: 5
5: 3
6: 6
7: -1
8: -1
9: -1
10: -1
11: -1
12: -1
```

Predecesores:

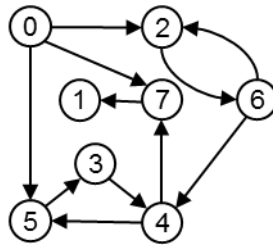
```
0: 0
1: 0
2: 0
3: 5
4: 3
5: 0
```

```

6: 4
7: -1
8: -1
9: -1
10: -1
11: -1
12: -1

```

Considere, en cambio, este grafo dirigido, el cual se proporciona en el archivo `ejemplo2.txt`:



El resultado de la llamada `DeepWeb ejemplo2.txt 0 dfs --arb` debería ser:

Todas las paginas son parte de la internet visible.

Arbol:

```

0-0 (raiz)
  0-2 (arco de camino)
    2-6 (arco de camino)
      6-2 (arco de subida)
      6-4 (arco de camino)
        4-5 (arco de camino)
          5-3 (arco de camino)
            3-4 (arco de subida)
        4-7 (arco de camino)
          7-1 (arco de camino)
  0-5 (arco cruzado)
  0-7 (arco cruzado)

```

Sin embargo, `DeepWeb ejemplo2.txt 0 bfs --trunc 2 --pred` debería ser:

4

Predecesores:

```

0: 0
1: 7
2: 0
3: 5
4: -1
5: 0
6: 2
7: 0

```

6 Entrega

Debe entregar, a través del Moodle antes del viernes, 26 de octubre a las 11:55 pm. un archivo comprimido que debe contener los archivos fuentes del código de su solución.

7 Evaluación

De ser escogido para evaluación, se asignarán:

- 3 puntos por código
 - 1 punto por adaptar BFS
 - 1 punto por adaptar DFS
 - 1 punto por su programa principal
- 7 puntos por ejecución
 - 2 puntos por identificar la componente conexa correctamente
 - 1 punto con DFS
 - 1 punto con BFS
 - 1 punto por truncar correctamente
 - 2 punto por imprimir el árbol correctamente
 - 1 punto por indentar correctamente
 - 1 punto por etiquetar las aristas
 - 1 punto por imprimir los ordinales correctamente
 - 1 punto por identificar los predecesores correctamente

Se asignará un punto adicional si el programa es capaz de detectar cuántas componentes conexas tiene el grafo. Debe agregar un quinto parámetro llamado `--comp` que indique si se deben buscar ciclos. Debe imprimir esa información al final