

PRÁCTICA 10

BFS

MATÚ HERNÁNDEZ DIANA

ROJO MATA DANIEL

■ algoritmoBFS

- **Entrada:** Una gráfica G de orden n y tamaño m tal que $G.vertices = \{v_1, v_2, \dots, v_n\}$ y $vi.identificador = vi$ y el identificador de un vértice $s \in G.vertices$.
- **Salida:** La gráfica G tal que el valor del campo visitado de los vértices alcanzados desde el vértice s es `true`.

```
1  public void algoritmoBFS(String s) throws Exception {
2      for (Vertice coso : this.vertices) {
3          Vertice v = darVertice(coso.identificador);
4          v.modificarVisitado(false);
5      }
6
7      Cola<String> colaVisitados = new Cola<>();
8      colaVisitados.encolar(s);
9
10     while (!colaVisitados.esVacia()) {
11         String u = colaVisitados.darElementoInicio();
12         colaVisitados.desencolar();
13         LinkedList<String> adj_u = darVecindad(u);
14
15         while (!adj_u.isEmpty()) {
16             String v = adj_u.getFirst();
17             adj_u.removeFirst();
```

```

18         Vertice w = darVertice(v);
19
20         if (!w.darVisitado()) {
21             colaVisitados.encolar(v);
22             w.modificarVisitado(true);
23         }
24     }
25     modificarVisitadoVertice(u, true);
26 }
27 }

```

Corrección del algoritmo

El algoritmo inicia con el ciclo `for` 2-5, quien es el encargado de recorrer todos los vértices de la gráfica para asegurarse de que todo vértice, v_i , cumpla que: $v_i.\text{identificador} = \text{false}$. Esto con el afán de no tener problemas en la ejecución del algoritmo.

Este ciclo se implementa de manera correcta, para demostrar esto, se proporciona un invariante de ciclo en forma de lema.

*LEMA: Después de la iteración i , el campo *visitado* de todos los vértices anteriores al vértice v_i , incluido éste, refleja la condición booleana *false*, esto es, $v_i.\text{visitado} = \text{false}, \forall j \in \{1, \dots, i\}$*

- **Inicialización:** El ciclo `for` itera sobre todos los vértices de la gráfica, lo que significa que la inicialización se lleva a cabo para cada uno de ellos. Después de la primera iteración ($i = 1$), se asegura que el campo `visitado` de v_1 sea establecido en `false`, pues se ejecuta la línea de código 4 en donde se cambia el estado de `visitado` a `false`. Este hecho garantiza que el invariante se cumpla después de la primer iteración del ciclo `for`, ya que la condición booleana requerida por el lema se satisface para el primer vértice.
- **Mantenimiento:** Supóngase que el vértice actual es v_i en la iteración i . Hasta este punto, la instrucción 4 se ha ejecutado en los vértices v_1, v_2, \dots, v_{i-1} , estableciendo sus campos

`visitado` en `false`. Luego, en la iteración actual, la instrucción 4 se ejecuta en v_i , asegurando que su campo `visitado` también se establezca en `false`. Por lo tanto, al final de la iteración i , todos los vértices anteriores o iguales a v_i (incluido v_i) cumplen el invariante, ya que todos tienen sus campos `visitado` en `false`.

- **Finalización:** Al recorrer todos los vértices, se ha aplicado la instrucción 4 a todos éstos, garantizando que el campo `visitado` de cada uno ha sido cambiado a `false`. Por lo que el invariante se cumple.

Ahora, se procede a dar un invariante para el ciclo `while` 10-25 en forma de lema.

LEMA: Antes de cada iteración del ciclo `while` exterior (líneas 10-25) y después de cada iteración del ciclo `while` interior (líneas 15-24), la cola `colaVisitados` contiene los vértices que deben ser procesados en el nivel actual del recorrido BFS. Además, los vértices en `colaVisitados` y sus vecinos han sido marcados como visitados.

- **Inicialización:** Antes de iniciar el ciclo exterior, se crea la cola `colaVisitados` y es hasta la línea 8 en donde se agrega el elemento `s` a ésta. Esto cumple con el hecho de que la cola mencionada contiene los vértices que deben ser procesados para el recorrido BFS, a saber, solo contiene a `s`.
- **Mantenimiento:** Durante cada iteración del bucle interior, se considera un vértice `u` que ha sido desencolado de `colaVisitados`. A continuación, se exploran todos los vecinos no visitados de `u` y se realizan las siguientes acciones:
 - Para cada vecino `v` de `u` que aún no ha sido visitado, se encola en `colaVisitados` y se marca como visitado. Esto se logra con la condición `if (!w.darVisitado())` (línea 20-23). Este paso garantiza que todos los vecinos no visitados de `u` se agregarán a la cola `colaVisitados` y se marcarán como visitados. Esto asegura que se cumple la segunda mitad del lema propuesto.
 - Después de este bucle interior, `colaVisitados` contiene los vértices que deben ser procesados en el próximo nivel del recorrido BFS. Además, los vecinos de `u` se han agregado a la cola y se han marcado como visitados. Finalmente, el marcado de los vértices garantiza el cumplimiento del invariante.

- **Finalización:** La condición de terminación establece que cuando la cola `colaVisitados` está vacía, todos los vértices alcanzables desde el vértice inicial `s` han sido visitados. La cola `colaVisitados` contiene los vértices que deben ser procesados en el nivel actual del recorrido, y cuando esta cola se vacía, significa que no hay más vértices por procesar en el nivel actual. Dado que se han marcado como visitados todos los vértices alcanzables desde el vértice inicial hasta el nivel actual, se puede concluir que el algoritmo ha explorado todos los vértices alcanzables en toda la gráfica. Esta propiedad además asegura la finalización del algoritmo de manera correcta después de haber visitado todos los vértices de la gráfica, esto es, el invariante se cumple.

Por lo argumentado anteriormente, se concluye que el algoritmo dado es correcto.