x

Juan Cisneros Resendiz
Algoritmos en gráficas 2021
Tarea 3
Fecha de entrega: 13 de febrero

El objetivo de esta tarea es que entiendas mejor qué es el orden topológico y las componentes fuertemente conexas de una gráfica dirigida. También, esta tarea deberá ayudarte a entender mejor el algoritmo de Kosaraju-Sharir y las ideas que sustentan que sea correcto.

1.

1. Describe un algoritmo que reciba como entrada una gráfica dirigida G y devuelva como salida $rev(G)$, la gráfica inversa. Tu algoritmo debe tener complejidad $O(V + E)$.
2. Demuestra que para toda gráfica dirigida G, la gráfica de componentes conexas $scc(G)$ es acíclica.
3. Demuestra que $ssc(rev(G)) = rev(scc(G))$ para toda gráfica dirigida G.

Respuesta:

1. La definición de $rev(G) = \{w \to v \mid v \to w \in G\}$ nos da un algoritmo para generar $rev(G)$.

```
REV(G):
    G' = una gráfica vacia
    for all vertices v in G
        for each edge v → w
            G' = {w → v} ∪ G'
    return G'
```

   Usando la representación de listas de adyacencia, la complejidad de construir una gráfica vacía G' es $O(1)$, y la complejidad de agregar una nueva arista a G' es $O(1)$ si agregamos v a la lista de adyacencia de w, así que la complejidad del algoritmo es $O(V + E)$.

   Este algoritmo es correcto, por cada arista en G de la forma $v \to w$ agregamos una a la gráfica invertida G' la arista $w \to v$. Nótese en particular que necesitamos ir guardando la gráfica invertida en otra gráfica sin modificar la original, ya que si en su lugar modificáramos G, entonces podría pasar que primero visitamos a un vértice v con una arista $v \to w$ y la invertimos, ahora tenemos la arista $w \to v$, y después visitamos w, ahora volveríamos a invertir la arista $w \to v$ y nos quedaríamos con la arista original.

2. Supongamos por contradicción que $scc(G)$ no es una DAG, entonces existe un ciclo dirigido C en la gráfica $scc(G)$. Sea $C = v_1, v_2, \dots v_k$, y sean $v_i \in C$, $v_j \in C$. Como v_i y v_j están en C existe un camino dirigido de v_i a v_j, y un camino dirigido de v_j a v_i. Es decir que $v_i \in reach(v_j)$ y $v_j \in reach(v_i)$.

   Llamemos S_i a la componente que v_i representa y llamemos S_j a la componente que v_j representa.

   Sabemos que como S_i es una componente fuertemente conexa, todo vértice $w \in S_i$ es alcanzable por cualquier otro vértice en S_i, de igual manera para S_j, así que para cualquier vértice $w \in S_i$ y cualquier vértice $x \in S_j$. Como hay un ciclo $C = v_1, v_2, \dots v_k$ en $scc(G)$, al ver cada componente fuertemente conexa S_m que es representada por v_m, hay una arista de G $b_m \to a_{m+1}$ tal que $b_m \in S_m$ y $a_{m+1} \in S_{m+1}$, y en el caso que $k = m$, hay una arista $b_k \to a_1$ tal que $b_k \in S_k$ y $a_1 \in S_1$, es decir, hay una arista que nos lleva de la componente S_i a la componente S_{i+1} (si no existiera esa arista entonces el vértice v_i que representa esa componente no estaría en el ciclo C).

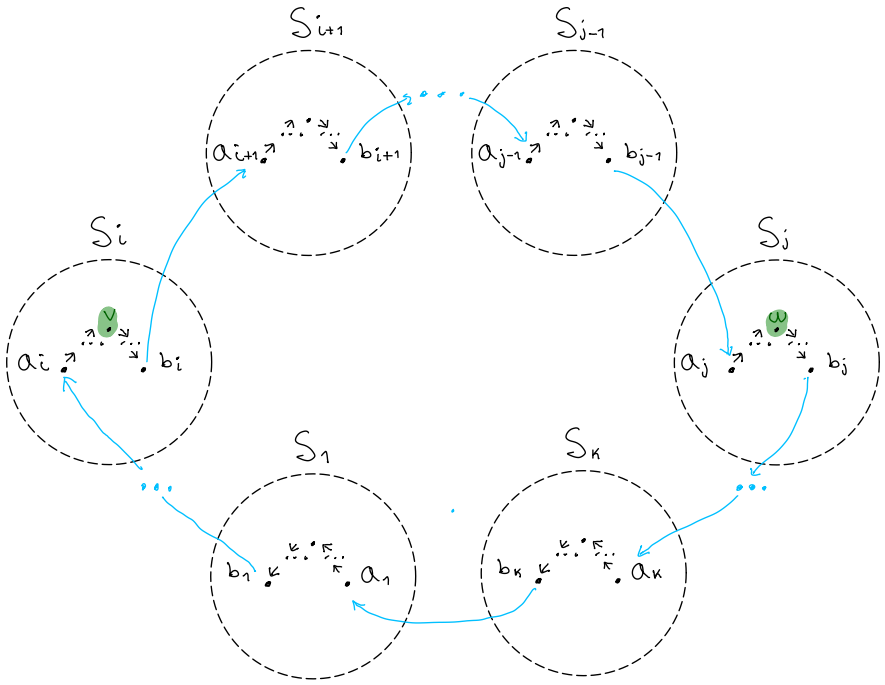


Figura 1: Ciclo C de la gráfica de componentes conexas

Sea v cualquier vértice en S_i y w cualquier vértice en S_j, podemos formar el camino de v a w $(v \to \ldots \to b_i) \to (a_{i+1} \to \ldots \to b_{i+1}) \to a_{i+2} \to \ldots \to b_{j-1} \to (a_j \to \ldots \to w)$ y el camino de w a v $(w \to \ldots \to b_j) \to a_{j+1} \to \ldots \to (a_k \to \ldots \to b_k) \to a_1 \to \ldots \to (a_i \to \ldots \to v)$, eso implica que v y w son alcanzables entre sí, eso implica que tanto S_i como S_j son una sola componente conexa, eso contradice que $scc(G)$ sea la gráfica de componentes.

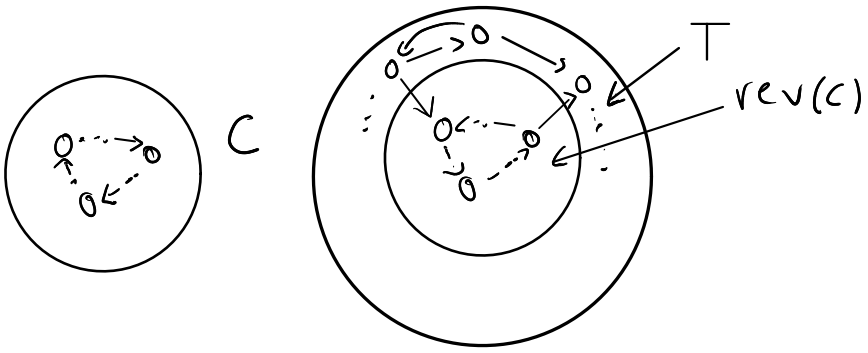
3. Sea C una componente conexa de G, entonces todo $v \in C$ alcanza a todo vértice $u \in C$. Al considerar la gráfica invertida $rev(G)$, llamemos $rev(C)$ a la inversa de la componente C. Como $v \in C$ alcanza a todo $v \in C$, y la gráfica inversa no agrega ni remueve vértices, entonces $rev(C)$ y C tienen los mismo vértices. Ahora queremos demostrar que $rev(C)$ forma una componente fuertemente conexa en la gráfica $rev(G)$.

   Supongamos por contradicción que $rev(C)$ no es una componente fuertemente conexa de $rev(G)$. Existen dos casos

   1) Los vértices de $rev(C)$ no forman una componente, si no que están en varias componentes conexas.
   2) Los vértices de $rev(C)$ están contenidos en una componente conexa T

   Para el caso 1) existen al menos dos vértices $v \in rev(C)$ y $w \in rev(C)$ tal que $v \in C_1$ y $w \in C_2$, donde C_1 y C_2 son componentes fuertemente conexas diferentes de $rev(G)$. Al considerar la gráfica G, hay un camino de v a w y de w a v ya que C es una componente conexa de G, entonces hay un camino de w a v y de v a w en la gráfica $rev(G)$, es decir, v y w se alcanzan entre sí, eso implica que están en la misma componente conexa, contradiciendo que C_1 y C_2 son componentes conexas diferentes.

   Para el caso 2) sea T la componente fuertemente conexa de $rev(G)$ en la que viven todos los vértices $v \in rev(C)$.



entonces $V(rev(C)) \subseteq V(T)$. Como T contiene a todos los vértices de $rev(C)$ y $rev(C)$ no es una componente fuertemente conexa, entonces existe un $w \in V(T)$ tal que $w \notin V(rev(C))$. Como w y todo $v \in V(rev(C))$ están en la componente fuertemente conexa T, para cualquier $w \in V(T)$ existe un camino desde w a v, y de v a w, entonces en la gráfica G esos caminos van de v a w y de w a v, y como $V(C) = V(rev(C))$, en la gráfica G w alcanza a todos los vértices de C, y todos los vértices de la componente fuertemente conexa C alcanzan a w, contradiciendo que C es una componente fuertemente conexa.

Esto implica que por cada componente fuertemente conexa C de la gráfica G, existe una componente fuertemente conexa $rev(C)$ de la gráfica $rev(G)$ que tiene los mismos vértices.

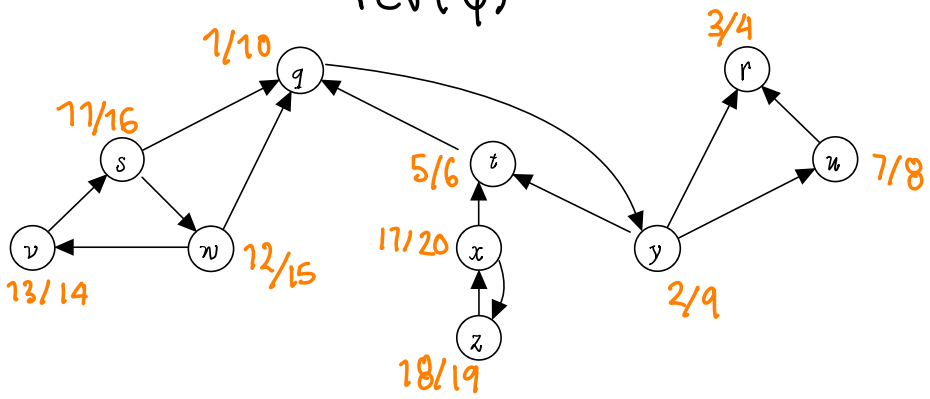
Por la definición de $rev(G)$, $scc(G)$ y $scc(rev(G))$ tienen los mismos vértices pero con las aristas con la dirección invertida, es decir, para cada arista $(v \to w) \in E(scc(G))$ tenemos a $(w \to v) \in E(scc(rev(G)))$. Tanto $scc(G)$ como $scc(rev(G))$ son gráficas acíclicas, entonces sus componentes conexas son sus mismos vértices, si invertimos las aristas en $scc(G)$ obtenemos, por cada arista $(v \to w) \in E(scc(G))$, a $(w \to v) \in E(rev(scc(G)))$, que son precisamente las mismas aristas que tiene $scc(rev(G))$, y tanto $scc(rev(G))$ como $rev(scc(G))$ tienen los mismos vértices, así que $rev(scc(G)) = scc(rev(G))$.

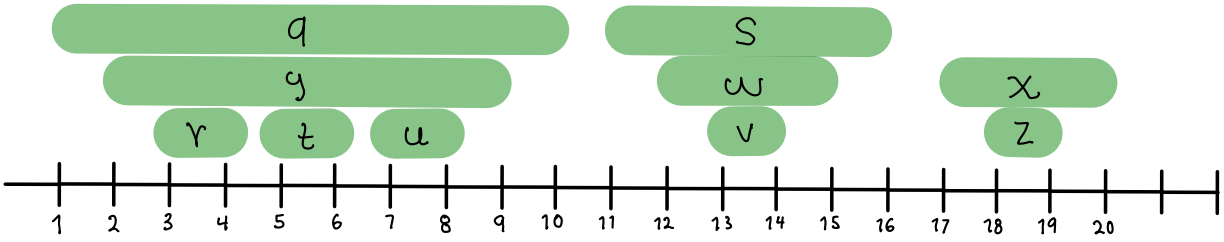
■

2. Muestra cómo el algoritmo de Kosaraju-Sharir trabaja en la gráfica de la Figura . Muestra la gráfica de intervalos que hemos hecho en clase, las componentes conexas resultantes, el stack que usa el algoritmo para calcular el post-orden, etc. Supón que DFS considera los vértices en orden alfabético y que las listas de adyacencia están en orden alfabético.

Fase 1

rev(G)



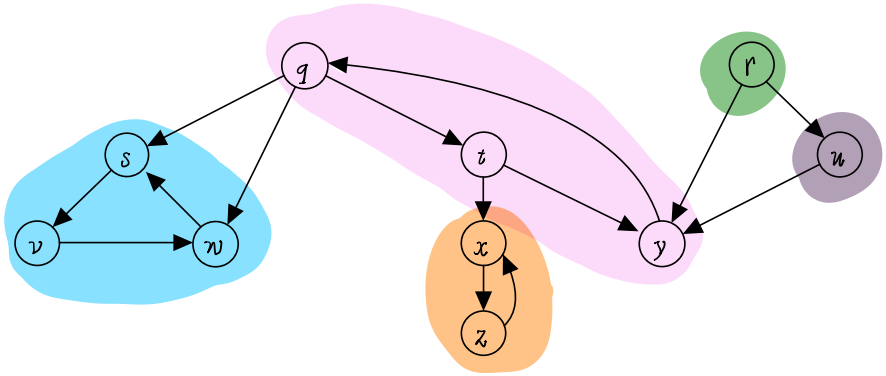


Stack

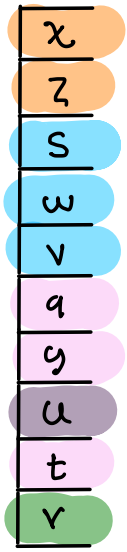
x
z
s
w
v
q
y
u
t
r

Fase 2

G



Stack



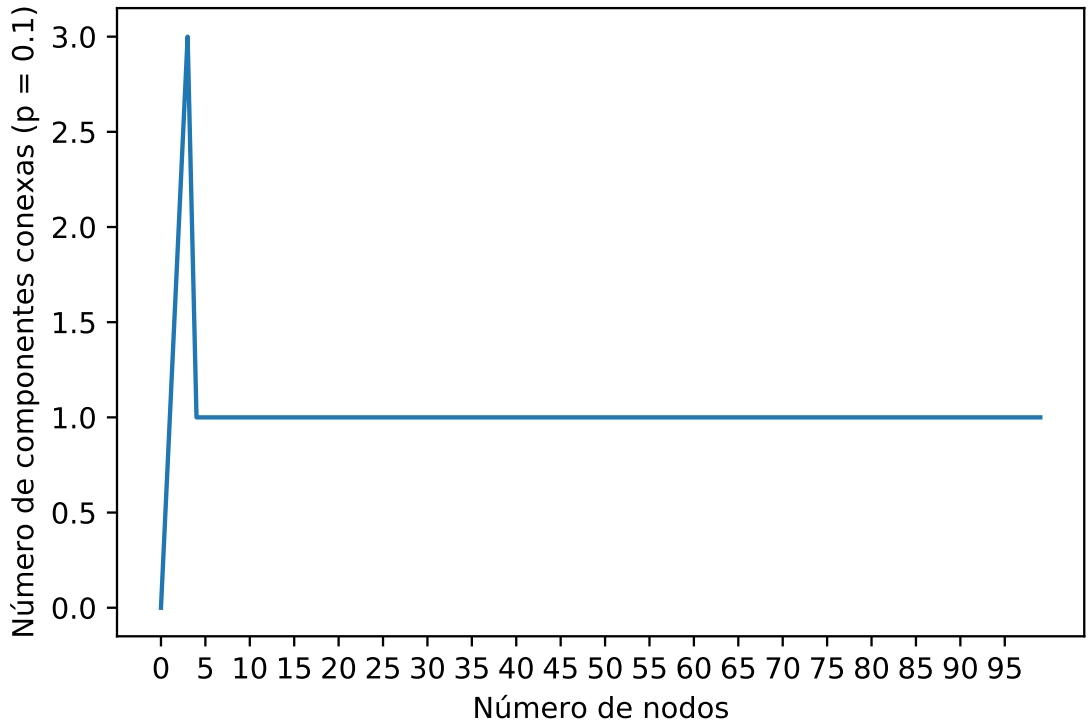
3. Para resolver este problema vas a usar Sage. Imagina el conjunto de todas las gráficas dirigidas con n vértices. Ahora imagina que calculas para cada una de estas gráficas su número de componentes fuertemente conexas. Elige el número que más se repita. ¿Cuánto crees que sea ese número? ¿Estará más cerca del uno o más cerca del n?

1. Fija n.
2. Genera una gráfica dirigida aleatoria usando Sage, deberás utilizar las funciones que Sage tiene para este fin. Mira la documentación.
3. Una vez que generes la gráfica calcula su número de componentes conexas, usando también la función que Sage tiene para este fin. Mira la documentación.
4. Realiza experimentos con un número suficientemente grande de gráficas generadas de manera aleatoria. Para cada gráfica almacena la cantidad de componentes conexas, con estos datos genera una curva graficando el número de vértices de la gráfica versus el número de componentes conexas.
5. Repite hasta obtener un número suficientemente grande de valores distintos de n.

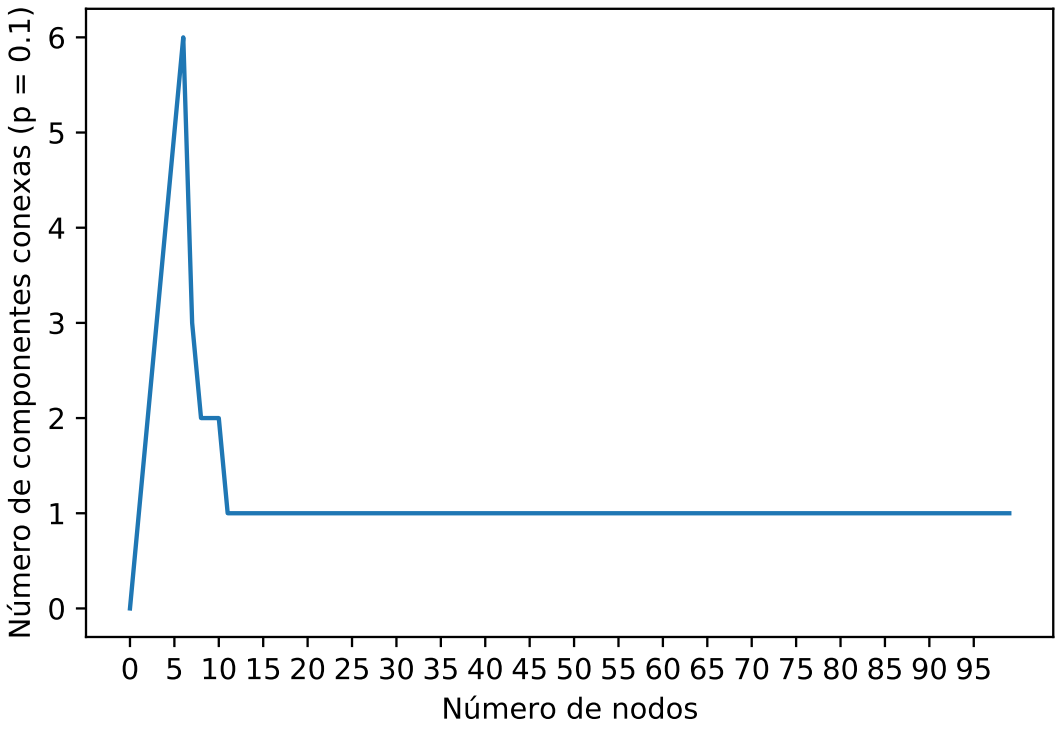
¿Qué puedes concluir de este experimento? Discute.

Para generar gráficas aleatorias se usó el método `digraphs.RandomDirectedGNP(n,p)`, que genera una digráfica aleatoria de tamaño `n` con probabilidad `p` de agregar una arista de un nodo v a un nodo u.

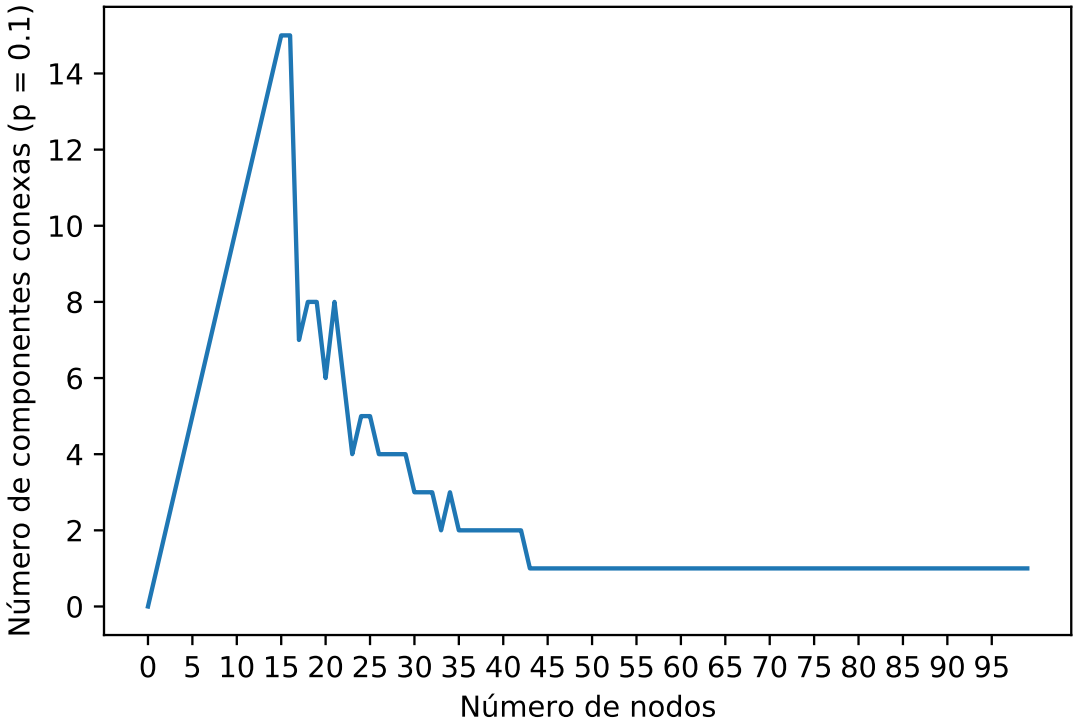
Se realizaron experimentos con gráficas de tamaño 1 a 100, y con valores de p 0.5 (50 % de probabilidad de que exista una arista), 0.25 y 0.1, en las gráficas se muestran el número de vértices de las gráficas y el número de componentes conexas que más se repite. En los tres experimentos mientras más grande se hacia la gráfica más se acercó a uno, yo pienso que esto pasa porque para generar una componente conexa de tamaño k solo necesitamos generar un ciclo entre k vértices, eso es más probable que suceda cuando n se vuelve más grande.



Experimento con $p = 0{,}5$



Experimento con $p = 0{,}25$



Experimento con $p = 0{,}01$

■