CS2023 - Aula de Ejercicios Nº 11

Brenner H. Ojeda Rios

Semestre 2024-0

Se sugiere que cada estudiante trate de resolver los ejercicios de forma **individual** y luego los discuta en grupo.

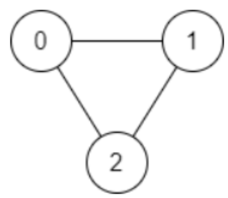
Ejercicios

1. (5 pts) Hay un grafo G con n vértices, donde cada vértice está etiquetado de 0 a $n-1$ (inclusive). Las aristas en G se representan como un arreglo 2D de enteros, donde cada $arista[i] = [u_i, v_i]$ denota una arista bidireccional entre el vértice u_i y el vértice v_i. Cada par de vértices está conectado como máximo por una arista y ningún vértice tiene un bucle.

   Quiere determinar si existe una ruta válida desde el origen del vértice hasta el destino del vértice.

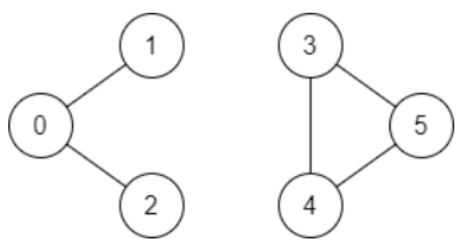
   Dadas las aristas y los números enteros n, origen y destino, devuelve verdadero si hay una ruta válida desde el origen al destino, o falso en caso contrario.

   - **Ejemplo 1:**

     `Input: n = 3, edges = [[0,1],[1,2],[2,0]], source = 0, destination = 2`
     `Output: true`
     `Explicación:` Hay dos caminos del vértice 0 para el vértice 2: (a) $0 \to 1 \to 2$, (b) $0 \to 2$.

   - **Ejemplo 2:**

     `Input: n = 6, edges = [[0,1],[0,2],[3,5],[5,4],[4,3]], source = 0, destination = 5`
     `Output: false`
     Explicación: No hay un camino del vértice 0 para el vértice 5.

   Restricciones:

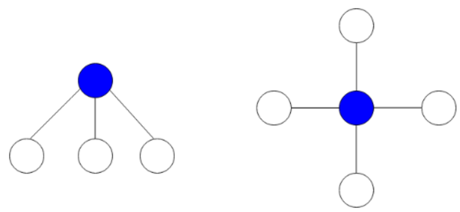
   - $1 \leq n \leq 2 * 10^5$
   - $0 \leq$ `edges.length` $\leq 2 * 10^5$
   - `edges[i].length` $= 2$
   - $0 \leq u_i, v_i \leq n-1$
   - $u_i \neq v_i$
   - $0 \leq$ `source, destination` $\leq n-1$
   - No hay aristas duplicadas.
   - No hay bucles.

2. Hay un grafo no dirigido que consta de n nodos numerados del 0 al $n-1$. Se le proporciona un arreglo `vals` de enteros (indexado en 0) de longitud n, donde `vals[i]` denota el valor del i-ésimo nodo.

   También se le proporciona un arreglo 2D de números enteros, donde $edges[i] = [a_i, b_i]$ denota que existe una arista no dirigida que conecta los nodos a_i y b_i.

   Un grafo estrella es un subgrafo del grafo dado que tiene un nodo central que contiene 0 o más vecinos. En otras palabras, es un subconjunto de aristas del grafo dado tal que existe un nodo común para todas las aristas.

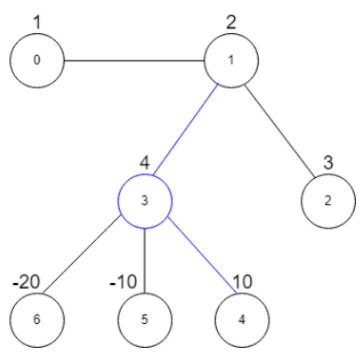
   La siguiente imagen muestra grafos estrellas con 3 y 4 vecinos respectivamente, centrados en el nodo azul.



   Denotamos con `suma estrella` a la suma de los valores de todos los nodos presentes en el grafo estrella.

   Dado un número entero k, devuelve la máxima `suma estrella` de un grafo de estrellas que contiene como máximo k aristas.

   - **Ejemplo 1:**



     `Input: vals = [1,2,3,4,10,-10,-20], edges = [[0,1],[1,2],[1,3],[3,4],[3,5],[3,6]], k = 22`

     `Output: 16`

     `Explicación:` El diagrama anterior representa el grafo de entrada. El grafo estrella con la máxima `suma estrella` se indica en azul. Está centrado en 3 e incluye a sus vecinos 1 y 4. Se puede demostrar que no es posible obtener un grafo estrella con una suma mayor que 16.

   - **Ejemplo 2:**

     `Input: vals = [-5], edges = [], k = 0`

     `Output: -5` (Sólo hay un grafo estrella posible, que es el propio nodo 0. Por tanto, devolvemos -5.)

   Restricciones:

   - n = `vals.length`
   - $1 \leq n \leq 10^5$
   - $-10^4 \leq$ `vals[i]` $\leq 10^4$
   - $0 \leq$ `edges.length` $\leq min(n*(n-1)/2, 10^5)$
   - `edges[i].length` = 2.
   - $0 \leq a_i, b_i \leq n-1$
   - $a_i \neq b_i$
   - $0 \leq k \leq n-1$

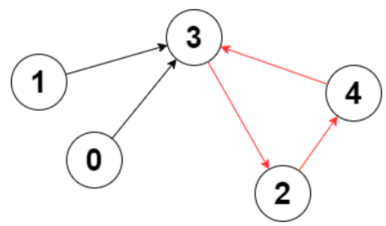
3. Se le proporciona un grafo dirigido de n nodos numerados del 0 al $n-1$, donde cada nodo tiene como máximo una arista saliente.

   El grafo se representa con un arreglo $edges$ 0-indexado de tamaño n, lo que indica que hay una arista dirigida desde el nodo i hasta el nodo $edges[i]$. Si no hay una arista de salida desde el nodo i, entonces $edges[i] =$ -1.

   Devuelve el tamaño del ciclo ***más largo*** *del grafo*. Si no existe ningún ciclo, devuelve -1.

   Un ciclo es un camino que comienza y termina en el **mismo** nodo.

   - **Ejemplo 1:**

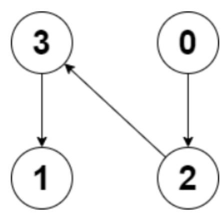
     

     `Input: edges = [3,3,4,2,3]`

     `Output: 3`

     `Explicación:` El ciclo más largo en el grafo es el ciclo: $2 \to 4 \to 3 \to 2$. La longitud de este ciclo es 3.

   - **Ejemplo 2:** `Input: edges = [2,-1,3,1]`

     `Output: -1`

     `Explicación:` No hay ciclos en este grafo.

   Restricciones:

   - $n =$ `edges.length`
   - $2 \leq n \leq 10^5$
   - $-1 \leq$ `edges[i]` $< n$
   - `edges[i]` $\neq i$