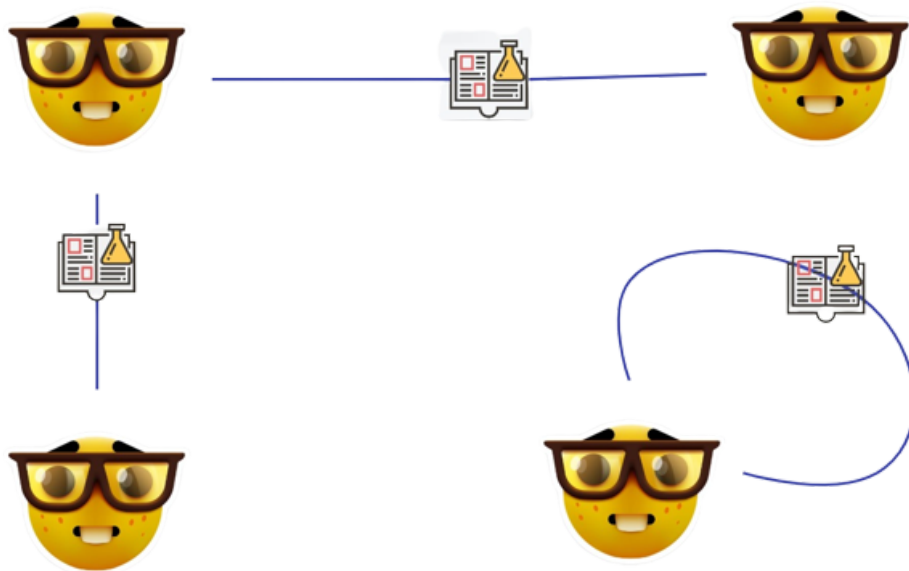


UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE CIENCIAS  
**Análisis de Algoritmos 2023-2**

- Dania Paula Góngora Ramírez  
- Diana Laura Salgado Tirado

## EJERCICIO SEMANAL 10

Hay una cantidad  $N$  científicos en la comunidad, algunos científicos están conectados entre sí y otros no. La conexión entre ellos es transitiva; si el científico A tiene una colaboración directa con el científico B, y el científico B tiene una colaboración directa con el científico C, entonces el científico A tiene una colaboración indirecta con el científico C, entonces los científicos A, B y C forman un grupo. Un grupo de científicos está definido como un conjunto de científicos que están conectados directa o indirectamente a través de colaboraciones. Si no hay ninguna colaboración registrada para un científico D, entonces ese científico no tiene grupo y forma un grupo individualmente en el que el grupo se compone por solo el científico D



- a) Calcula cuántos grupos hay para G científicos que se relacionan de la siguiente forma:  
A con D, C con B y F con G

Se pueden formar 8, ya que los grupos existentes son (A, D), (C, B), (F, G). Los nuevos 8 grupos que se pueden formar al considerar un miembro de cada grupo son (A, C, F), (A, C, G), (A, B, F), (A, B, G), (D, C, F), (D, C, G), (D, B, F) y (D, B, G).

- a) Haz un algoritmo que calcule para n cantidad de científicos.

En resumen tenemos que contar las componentes conexas de una gráfica no dirigida, esto lo haremos utilizando DFS, cada que encontremos un vértice no visitado, esto nos indica que hemos encontrado un nuevo grupo. El número total de grupos será igual al número de veces que comenzamos una búsqueda desde un vértice no visitado.

Algoritmo:

```
#Inicializamos todos los vértices como no visitados
visitado = [0]

grupos_existentes = 0
nuevos_grupos = 1

por cada vertice:

    if (visitado[i] == Falso):
        grupos_existentes += 1

        # Número de nuevos grupos que se pueden formar.
        nuevos_grupos = (nuevos_grupos * NoVisitados(i, visitado))

    if (grupos_existentes == 1):
        nuevos_grupos = 0
```

Función auxiliar NoVisitados:

```
contador = 1
visitado[v] = Verdadero
i = 0
mientras iteramos sobre los vértices adyacentes a v:
    if (no visitado el vertice adyacente i):
        contador = contador + NoVisitados(v_i, visitado)
        i += 1
return contador
```

