

## Programa 01

Profesora: María de Luz Gasca Soto

Ayudante: Brenda Margarita Becerra Ruíz

Ayudante: Malinali González Lara

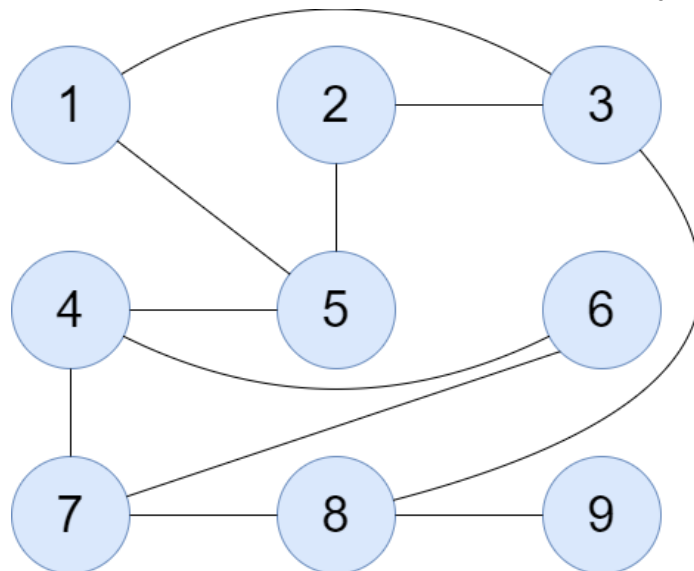
26 de febrero de 2024

### Actividades.

1. Considera los siguientes problemas de decisión:
  - a. Ruta más corta: Dada una gráfica no dirigida  $G = (V, E)$ ,  $u, v$  vértices en  $V$ , y  $k$  entero positivo, ¿ ¿Existe una  $uv$ -trayectoria en  $G$  de peso menor a  $k$ ?
  - b. 3-SAT.
2. Para cada uno de estos problema de decisión deberás:
  - a. Dar su forma canónica.
  - b. Diseñar un algoritmo no-determinístico polinomial.  
Recuerda que diseñar un algoritmo **no** significa dar código
  - c. Implementar el algoritmo diseñado

### Indicaciones.

1. Deberás elaborar un archivo PDF con las actividades 2a y 2b.
2. Cada uno de tus programas deberá:
  - a. Tener las dos fases de todo algoritmo no-determinístico.
  - b. Mostrar el ejemplar de entrada en pantalla.
    - i. **Gráfica:** El ejemplar de **entrada** debe ser brindado por el usuario a través de un archivo de texto (.txt) con el siguiente formato: Primer línea indicarán los vértices de la gráfica separados por una coma; de la segunda línea en adelante indicará las aristas de la gráfica, una por línea, como un par de vértices separados por una coma. Ejemplo:



El ejemplar representante de la gráfica anterior sería el siguiente:

---

```

1,2,3,4,5,6,7,8,9
2,3
4,5
8,9
3,1
5,1
7,8
6,4
8,3
6,7
2,5
7,4|

```

Dicho ejemplar deberá ser tal que si  $|V| = n$  y  $|A| = m$ , entonces  $10 \leq n \leq 20$ , y  $m > 2n$ . Cualquier otro parámetro del ejemplar de entrada, podrá ser fijo o generado de manera aleatoria, esto deben especificarlo en su diseño.

- ii. **3-SAT**: El ejemplar de **entrada** deberá tener entre 3 y 10 cláusulas. De igual forma, deberá ser brindado por medio de un archivo de texto (.txt) con el siguiente formato:

$$(x \vee \neg y \vee z) \wedge (x \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z)$$

Dada la fórmula booleana anterior, su ejemplar correspondiente sería:

---


$$(x + -y + z) * (x + y + z) * (-x + -y + -z) * (-x + y + -z)|$$

- c. Para la **salida** de ambos problemas se deberá mostrar en pantalla la salida de la **fase adivinadora** de tu algoritmo para dicho ejemplar así como la solución de la **fase verificadora** final.
- Deberás enviar las instrucciones para compilar y ejecutar tu programa en un archivo README.txt.
  - Deberás enviar tus implementaciones en un archivo **.zip** con el nombre **[ApaternoAmaternoNombres]P01** a través del classroom.
  - FECHA LÍMITE DE ENTREGA: 9 de Marzo de 2024, 22:59.**