Identificación:

Sección: 1

Grupo Sicua+: 08

Nombre de autores: Laura Ávila, Sebastián Valencia

Identificación autores: 201212736, 201111578

1. **Algoritmo de solución:**
   El algoritmo inicia con la organización de los nodos que ingresan por parámetro en una matriz booleana. Aunque la matriz que se construye es de n+1xn+1 solo se usara hasta la diagonal izquierda. Para ilustrar lo anterior se muestra la matriz para el ejemplo que se plantea en el enunciado:

   Ubicación de nodos en la matriz

012	**102**		
	111		
	120		**300**

   Matriz booleana (esta es la que se construye)

true	**True**		
	True		
	true		**true**

   Si todos los nodos fueran negros, es decir, entraran por parámetro, la zona azul tendría números. Con esta matriz, la primer columna contiene el lado x, la diagonal sombreada el lado y, y la última fila el lado z.

   Para empezar el algoritmo verifica si hay true en la primera columna, si no hay no puede ganar el negro ya que no habría lado X y la respuesta es B. Si existe un true este nodo tiene conexión con los siguientes nodos:

true	**True**		
	True		
	true		**true**

   En el juego estos son el nodo a la derecha en el mismo nivel y los dos nodos en el nivel inferior con los que está conectado. Recursivamente se visitan los nodos hijos verificando si son true, si están en la diagonal o si están en la última fila. En el ejemplo anterior, nunca se llega al nodo 300 (esquina inferior derecha) dado que sus “padres” no son true.

Existen dos variables booleanas Y y Z, cuando se verifica que un hijo sea true y esté en la diagonal sombreada, la variable Y se pone true, es decir, ya encontramos el lado Y. Cuando un hijo sea true y este en la última fila, Z es true y encontramos el lado Z. Cuando Y y Z son true el negro gana, dado que encontró el lado X para poder iniciar el método recursivo y en sus hijos encontró el lado Y y el lado Z.

2. **Análisis de complejidades temporal y espacial:**
   a. **Complejidad espacial**
      Se crea al inicio una matriz booleana con las características que ya se mencionaron anteriormente, esta matriz tiene dimensiones (n+1) x (n+1). Por lo tanto:

$$S_n = \theta\big((n+1)x(n+1)\big)$$
$$S_n = O(n^2)$$

   b. **Complejidad temporal**
      Primero se recorre la matriz por la primera columna, en el peor de los casos se llega a la última posición con n+1 casillas. Cuando encuentra un true visita a los hijos true de esta casilla, en el peor de los casos m nodos, si todos los nodos que ingresan por parámetro están conectados. Entonces:

$$S_n = \theta\big((n+1)x(m)\big)$$
$$S_n = O(n * m)$$

3. **Cometarios finales**
   La complejidad espacial podrá hacerse menor debido a que estamos creando una matriz de (n+1)x(n+1) pero podría crearse con solo $(n^2+n)/2$ dado que solo se usa hasta la diagonal de la matriz.