**Estructuras de Datos y Algoritmos – iic2133**

**Control 4**

6 de mayo, 2019

**Nombre**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1**) Un *puente* en un grafo no direccional es una arista (*u*, *v*) tal que al sacarla del grafo hace que el grafo quede desconectado —o, más precisamente, aumenta el número de componentes conec-tadas del grafo; en otras palabras, la única forma para ir de *u* a *v* en el grafo es a través de la arista (*u*, *v*). [**4 pts.**] Explica cómo usar el algoritmo DFS para encontrar eficientemente los puentes de un grafo no direccional; y [**2 pts.**] justifica qué tan eficientemente. Recuerda que DFS asigna tiempos de descubrimiento (y finalización) a cada vértice que visita.

**2**) Sea un grafo no direccional y un subconjunto de sus vértices. Se dice que es un *-clique* si y todo vértice de está conectado con todos los otros vértices de .

Dado un grafo cualquiera y un número , queremos determinar si existe un -*clique* dentro de . Esto se puede resolver usando *backtracking*.

**a**) Describe la modelación requerida para aplicar *backtracking* a este problema: explica cuál es el conjunto de **variables**, cuáles son sus **dominios**, y describe en palabras cuáles son las **restricciones** sobre los valores que pueden tomar dichas variables.

**b**) Propón una **poda** y explica la modelación a nivel de código requerida para implementarla eficientemente.

**c**) Propón una **heurística para el orden de las variables** y explica la modelación a nivel de código requerida para implementarla eficientemente.