

Complejidad Computacional

Tarea 1

Karla Adriana Esquivel Ramírez
Andrea Itzel González Vargas
Luis Pablo Mayo Vega
Carlos Gerardo Acosta Hernández

Entrega: 06/03/17
Facultad de Ciencias UNAM

Ejercicios

- 1.
- 2.

Demuestra que los siguientes problemas sobre gráficas (es decir, los lenguajes respectivos) están en P (elige la representación que prefieras para las gráficas):

- CONNECTED: el conjunto de todas las gráficas conexas.

Por el ejemplo de la pagina 25 del libro Arora sabemos que decidir si dos vértices son conexos toma tiempo polinomial. Por lo que podemos esbosar un algoritmo para decidir la coneccidad de una grafica de la siguiente manera:

Checamos para cada par de vértices conectados por una arista dentro de G si son conexos. Si esto se cumple entonces la gráfica es Conexa y está en P. Esta en P por que a lo mas checamos $n \cdot n$ pares de vertices y cada ves que checamos nos toma tiempo polinomial (por lo mencionado anteriormente), entonces el tiempo que tarda nuestro algoritmo es polinomial.

- BIPARTITE: el conjunto de todas las gráficas bipartitas, es decir, aquellas cuyos vértices puedan ser divididos en dos conjuntos A y B tales que todas las aristas en la gráfica tengan en un extremo un vértice de A y en el otro uno de B.

Sea G una gráfica.

Tomamos todos los vértices que no tengan adyacencias entre sí y los colocamos en un mismo conjunto al que denotaremos A. Este paso toma a lo mas (n^2) .

Checamos los vértices sobrantes y comprobamos que no haya adyacencias entre ellos si hay alguna adyacencia entre ellos entonces no hacemos nada pues G no es bipartita, si no hay adyacencias entonces los colocamos en un segundo conjunto que denotaremos como B , si por alguna razón no sobraron vértices entonces simplemente no hacemos nada porque entonces G no es bipartita. Este paso toma a lo mas (n^2) pues revisas por segunda vez los nodos para checar las adyacencias.

Una vez que tengamos nuestros conjuntos A y B checamos que en cada uno de los vértices exista al menos una adyacencia que vaya de cada uno de los vértices del conjunto A a los vértices del conjunto B y viceversa. Este paso toma a lo mas (n^2) pues se tienen que revisar las adyacencias entre cada uno de los vértices dentro de nuestros conjuntos A y B .

Este algoritmo nos tomaria a lo mas $3(n^2)$ y esto es tiempo polinomial.

3.

Pablo

4.

Demuestra que los siguientes lenguajes están en NP:

- $2COL = \{ \langle G \rangle \mid \langle G \rangle \text{ codifica una gráfica que tiene una 2-coloración} \}$.

Probar que $2COL \in NP$.

Entrada para $TM = \langle A, w, c \rangle$.

Donde $A = G$ y G es una gráfica.

c es Una Lista de nodos coloreados.

w se describe de la siguiente manera.

1. Se checa que todos los nodos de G se encuentran en la Lista.
2. Se checa que todos los nodos de la Lista se encuentren en G .
3. Checar que para cualesquiera dos nodos conectados por una arista (adyacentes) serán de colores distintos, respectivamente $C1$ y $C2$.
4. Acepta si 1,2 y 3 se cumplen.

- $3COL = \{ \langle G \rangle \mid \langle G \rangle \text{ codifica una gráfica que tiene una 3-coloración} \}$.

Probar que $3COL \in NP$.

Entrada para $TM = \langle A, w, c \rangle$.

Donde $A = G$ y G es una gráfica.

c es Una Lista de nodos coloreados.

w se describe de la siguiente manera.

1. Se checa que todos los nodos de G se encuentran en la Lista.
 2. Se checa que todos los nodos de la Lista se encuentren en G .
 3. Checar que para cualesquiera dos nodos conectados por una arista (adyacentes) serán de colores distintos, respectivamente $C1$ y $C2$.
 4. Acepta si 1,2 y 3 se cumplen.
- ¿Cuáles están en P ?
 $2COL \in P$ porque una grafica es 2-coloreable si es una gráfica bipartita y por el ejercicio 2 sabemos que esto esta en P .
 $3COL \in NP\text{-Duro}$ según la literatura.