

Examen parcial – 27/05/2024

Para todo el parcial, damos la definición de *Dominating Set*:

Un set dominante (Dominating Set) de un grafo  $G$  es un subconjunto  $D$  de vértices de  $G$ , tal que todo vértice de  $G$  pertenece a  $D$  o es adyacente a un vértice en  $D$ . El problema de decisión del set dominante implica, dado un grafo  $G$  y un número  $k$ , determinar si existe un set dominante de a lo sumo tamaño  $k$ .

1. El *Hitting Set Problem* es: Dado un conjunto de elemento  $A$  de  $n$  elementos,  $m$  subconjuntos  $B_1, B_2, \dots, B_m$  de  $A$  ( $B_i \subseteq A \forall i$ ), y un número  $k$ , ¿existe un subconjunto  $C \subseteq A$  con  $|C| \leq k$  tal que  $C$  tenga al menos un elemento de cada  $B_i$  (es decir,  $C \cap B_i \neq \emptyset$ )?

*Dominating Set* es un problema NP-Completo. Demostrar que *Hitting Set Problem* es un problema NP-Completo, utilizando *Dominating-Set* para esto.

2. Implementar un algoritmo greedy que permita obtener el Dominating Set mínimo (es decir, que contenga la menor cantidad de vértices) para el caso de un árbol (en el contexto de teoría de grafos, no un árbol binario). Indicar y justificar la complejidad del algoritmo implementado. Justificar por qué se trata de un algoritmo greedy. Indicar si el algoritmo siempre da solución óptima. Si lo es, explicar detalladamente, sino dar un contraejemplo.
3. Realizar un modelo de programación lineal que obtenga el mínimo Dominating Set de un Grafo no dirigido. En dicho grafo, cada vértice tiene un valor (positivo), y se quiere que dicho Dominating Set sea el de mínima suma de dichos valores.
4. Implementar un algoritmo que (por backtracking) dado un grafo no dirigido en el que sus vértices tienen valores positivos, permita obtener el Dominating Set de suma mínima. Es decir, aquel dominating set en el cual la suma de todos los valores de los vértices sea mínima (no es importante que la cantidad de vértices del set sea mínima). Por simplicidad, considerar que el grafo es conexo.
5. Sea  $G$  un grafo dirigido “camino” (las aristas son de la forma  $(v_i, v_{i-1})$ ). Cada vertice tiene un valor (positivo). Implementar un algoritmo que, utilizando **programación dinámica**, obtenga el Dominating Set de suma mínima dentro de un grafo de dichas características. Dar la ecuación de recurrencia correspondiente al problema. Indicar y justificar la complejidad del algoritmo implementado. Indicar y justificar la complejidad espacial del algoritmo implementado, y si hay una optimización que permita consumir menos espacio.

Examen parcial – 27/05/2024

Para todo el parcial, damos la definición de *Dominating Set*:

Un set dominante (Dominating Set) de un grafo  $G$  es un subconjunto  $D$  de vértices de  $G$ , tal que todo vértice de  $G$  pertenece a  $D$  o es adyacente a un vértice en  $D$ . El problema de decisión del set dominante implica, dado un grafo  $G$  y un número  $k$ , determinar si existe un set dominante de a lo sumo tamaño  $k$ .

1. El *Hitting Set Problem* es: Dado un conjunto de elemento  $A$  de  $n$  elementos,  $m$  subconjuntos  $B_1, B_2, \dots, B_m$  de  $A$  ( $B_i \subseteq A \forall i$ ), y un número  $k$ , ¿existe un subconjunto  $C \subseteq A$  con  $|C| \leq k$  tal que  $C$  tenga al menos un elemento de cada  $B_i$  (es decir,  $C \cap B_i \neq \emptyset$ )?

*Dominating Set* es un problema NP-Completo. Demostrar que *Hitting Set Problem* es un problema NP-Completo, utilizando *Dominating-Set* para esto.

2. Implementar un algoritmo greedy que permita obtener el Dominating Set mínimo (es decir, que contenga la menor cantidad de vértices) para el caso de un árbol (en el contexto de teoría de grafos, no un árbol binario). Indicar y justificar la complejidad del algoritmo implementado. Justificar por qué se trata de un algoritmo greedy. Indicar si el algoritmo siempre da solución óptima. Si lo es, explicar detalladamente, sino dar un contraejemplo.
3. Realizar un modelo de programación lineal que obtenga el mínimo Dominating Set de un Grafo no dirigido. En dicho grafo, cada vértice tiene un valor (positivo), y se quiere que dicho Dominating Set sea el de mínima suma de dichos valores.
4. Implementar un algoritmo que (por backtracking) dado un grafo no dirigido en el que sus vértices tienen valores positivos, permita obtener el Dominating Set de suma mínima. Es decir, aquel dominating set en el cual la suma de todos los valores de los vértices sea mínima (no es importante que la cantidad de vértices del set sea mínima). Por simplicidad, considerar que el grafo es conexo.
5. Sea  $G$  un grafo dirigido “camino” (las aristas son de la forma  $(v_i, v_{i-1})$ ). Cada vertice tiene un valor (positivo). Implementar un algoritmo que, utilizando **programación dinámica**, obtenga el Dominating Set de suma mínima dentro de un grafo de dichas características. Dar la ecuación de recurrencia correspondiente al problema. Indicar y justificar la complejidad del algoritmo implementado. Indicar y justificar la complejidad espacial del algoritmo implementado, y si hay una optimización que permita consumir menos espacio.