

Algoritmos

TAREA 3

Compilación

4 de diciembre de 2013

Considera el siguiente algoritmo: La entrada es una gráfica no dirigida G con pesos en las aristas. Inicialmente, por cada vértice v en G se construye un árbol que consiste únicamente de v . Mientras el número de árboles mantenidos en el algoritmo sea mayor a uno, toma un árbol T cualquiera y elige la arista de peso mínimo con un solo extremo en T , añadiéndola al bosque, y disminuyendo el número total de árboles mantenidos en el algoritmo en uno.

Demuestra que este algoritmo encuentra el árbol generador de peso mínimo (AGM) o presenta un contraejemplo.

Corrección:

Observaciones

1. Si existe una arista entre dos árboles del bosque será la de menor peso que pueda unirlos.
2. El grafo es conexo, lo cual asegura que el algoritmo termina, ya que en cada iteración se reduce el número de árboles en 1.

Para el algoritmo propuesto se tiene la siguiente invariante:

“El bosque está compuesto por dos sub-grafos que son árboles generadores de peso mínimo”.

Al principio es cierto ya que cada vértice es un subgráfico de G donde él mismo es su árbol generador de peso mínimo.

En cada ejecución del ciclo mientras, se selecciona la arista de salida de peso menor de un árbol y se une con el resto, cumpliendo la propiedad de corte, la cual dice que la arista que una dos partes de un árbol generador mínimo debe ser la de menor peso (demostración en clase), por lo cual la invariante se mantiene.

Al final de la ejecución, hemos unido todos los árboles en uno, que es un árbol generador mínimo, manteniendo la invariante.

En caso positivo, presenta una breve discusión sobre las diferencias (si las hay) de este algoritmo comparado con el algoritmo de Prim y el de Kruskal.

Prim y Kruskal son casos particulares de este algoritmo

Cuál es la complejidad de este algoritmo?

En este algoritmo se realizan dos operaciones principalmente:

1. La unión de los subárboles. Haciendo esta operación con la estructura de datos union /find optimizada, para una gráfica de n vértices con $n - 1$ uniones, la complejidad es $O((2n-1)\log^* n) = O(n\log^* n)$.

2. La otra operación que realiza el algoritmo es la búsqueda del elemento mas pequeño que el árbol pueda unir. Esto es un poco mas complejo, ya que se deberán realizar en el peor caso $1 + 2 + 3 + \dots + n - 1$ comparaciones, esto es $\frac{n(n-1)}{2} = O(n^2)$.
- Por lo tanto, el algoritmo es de orden $O(n^2)$.