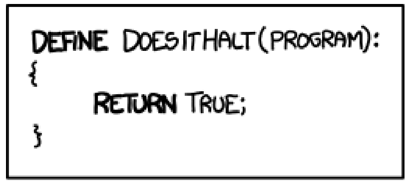
Segundo Certamen
Algoritmos y Complejidad

19 de enero de 2016



THE BIG PICTURE SOLUTION
TO THE HALTING PROBLEM

1. En su publicación original, Kruskal propuso otro algoritmo para hallar un árbol recubridor mínimo (*minimal spanning tree* en inglés), conocido como *reverse delete*: Dado un grafo conexo, sucesivamente se elimina un arco de mayor costo que no desconecta el grafo. Demuestre que este algoritmo es correcto.

   (40 puntos)

2. Dado un arreglo de n números a_1 a a_n, podemos usar el siguiente algoritmo para hallar el largo L de la subsecuencia creciente (no necesariamente consecutiva) más larga. Sea L_j el largo de la secuencia buscada que termina en a_j, podemos decir:

   $$L = \max_{1 \le j \le n} \{L_j\}$$
   $$L_j = \max_{\substack{i<j \\ a_i \le a_j}} \{L_i\}$$

   Complete el algoritmo basado en las relaciones esbozadas, y demuestre que es correcto.

   (40 puntos)

3. Considere un grafo cuyos vértices se rotulan con los números 1 a n, mientras los arcos se rotulan con la diferencia absoluta entre los rótulos de los vértices. Una rotulación se dice *graciosa* si los rótulos de los arcos son los números 1 a $n-1$, sin repeticiones. Un ejemplo para un camino de largo 5, P_5, es la secuencia $2, 5, 1, 3, 4$.

   Proponga un algoritmo para hallar todas las rotulaciones graciosas de P_n, basándose en generar las permutaciones de $1, \dots, n$ partiendo con un elemento e ir agregando alguno de los aún no incorporados. Explique cómo evita revisar todas las permutaciones.

   (35 puntos)