Diseño y análisis de algoritmos 2023-1

Tarea 6

Fecha de entrega: 22 de noviembre de 2023.

1. Tarea

1. 5pt. El problema de, dada una gráfica dirigida, $G = (V, A)$ y dos vértices s y t de la misma, determinar la longitud de la trayectoria más larga de s a t es un problema en general muy difícil. Pero si la gráfica es suficientemente simple, podemos resolver el problema de forma eficiente. En particular podemos si la gráfica pertenece a la siguiente familia.

   Una gráfica dirigida es *ordenable* si sus vértices se pueden indexar de tal forma, que toda flecha de la gráfica apunta de un vértice de índice menor a uno de índice mayor (es decir, para todos $i < j$, si v_i y v_j se conectan por una flecha, esta vá de i a j, y no al revés.

   Considera el problema de, dada una gráfica dirigida $G = (V, A)$ ordenable de n vértices: $v_1, \ldots, v_n$, *ya ordenada* (es decir, la etiquetación de los vértices ya cumple la propiedad de que toda flecha va de un vértice a otro con índice mayor), calcular la tráyectoria más larga del vértice v_1 al vértice v_n. Dá una solución de complejidad $O(|V| + |A|)$ para este problema, usando programación dinámica.

   a) 2pt. Propón la ecuación de Bellman del problema, demostrando que es correcta por inducción.

   b) 1pt. Propón la versión recursiva con memorización del algoritmo que resulta de aplicar la ecuación de Bellman. Analiza su corrección y complejidad.

   c) 1pt. Propón la versión iterativa de programación dinámica del algoritmo anterior. Analiza su corrección y complejidad.

   d) 1pt. Propón el algoritmo que, a partir de la(s) tabla(s) generadas por el algoritmo anterior, calcula una trayectoria de longitud máxima (los algoritmos anteriores calculan sólo su longitud).

2. 5pt. Dada una secuencia de números $S = s_1, s_2, \ldots, s_n$, una *subsecuencia creciente* es una subsecuencia de la original, con la propiedad de que sus elementos aumentan

monótonamente de forma estricta en valor (es decir, todo elemento es mayor que los previos a el). P. Ej. Si $S = 4, 2, -1, 3, 2, 3$, algunas de sus subsecuencias crecientes son: la secuencia 4; la secuencia $2, 3$ y la secuencia $-1, 2, 3$.

Considera el problema de, dada una secuencia de entrada $S = s_1, s_2, \ldots, s_n$, calcular alguna subsecuencia creciente de longitud máxima posible. En el ejemplo anterior, la subsecuencia $-1, 2, 3$ sería una respuesta válida (para ese ejemplo la subsecuencia de longitud máxima es única, pero en general no tiene porqué serlo). Dá una solución de complejidad $O(n^2)$ para este problema, usando programación dinámica.

Tip: considera la familia de subproblemas $\{\mathcal{I}_i(1 \leq i \leq n\}$ tal que $\mathcal{I}_i$ es el problema de determinar la longitud de la subsecuencia creciente más larga de S, tal que el s_i es el último elemento de esa secuencia.

a) 2pt. Propón la ecuación de Bellman del problema, demostrando que es correcta por inducción.

b) 1pt. Propón la versión recursiva con memorización del algoritmo que resulta de aplicar la ecuación de Bellman. Analiza su corrección y complejidad.

c) 1pt. Propón la versión iterativa de programación dinámica del algoritmo anterior. Analiza su corrección y complejidad.

d) 1pt. Propón el algoritmo que, a partir de la(s) tabla(s) generadas por el algoritmo anterior, calcula una subsecuencia creciente de longitud máxima (los algoritmos anteriores calculan sólo su longitud).