Algorísmia QP 2014–2015

Examen final **12 de Juny de 2015**

Duración: 2h 30m

Instrucciones generales:

- El ejercicio 1 es obligatorio
- Debéis escoger dos de entre los ejercicios 2, 3 y 4. Debéis indicar claramente cuáles son los ejercicios elegidos. Si resolvéis los tres ejercicios y la nota de los dos elegidos supera 4 puntos de los 6 posibles, se corregirá el tercer ejercicio y se sumará su nota a la nota de las otras preguntas contestadas.
- En cualquier caso, la calificación global de esta prueba escrita no podrá superar 10, al sumarse todos los puntos. Por ello, si la nota del apartado anterior supera a 10, la calificación global de la prueba escrita será 10.
- Para cada algoritmo que tengáis que desarrollar, garantizad que el algoritmo sea correcto (proporcione el tipo de solución requerida) y eficiente. Tenéis que proporcionar una descripción a alto nivel del algoritmo en pseudo-código, junto con las explicaciones y aclaraciones oportunas, de manera que se proporcionen: los argumentos que garantizan la corrección del algoritmo propuesto y el coste asintótico en caso peor del algoritmo (en función del tamaño de la entrada).
- Podéis hacer llamadas a algoritmos vistos en clase, pero si la solución es una variación de un algoritmo visto en clase tenéis que detallarlo.
- Se valorará especialmente la claridad del código (o pseudo-código) de las soluciones propuestas así como la de las explicaciones.
- Entregad por separado cada uno de los ejercicios y no olvidéis **marcar** en la hoja del problema 1 los dos problemas seleccionados (en el caso en que entreguéis los tres).

- **Ejercicio 1 (4 puntos)**
  Responder a esta pregunta en la hoja en blanco con espacio para marcar los problemas seleccionadas. Vuestra respuesta no puede utilizar más espacio y el resultado debe de ser legible. Justificad vuestras respuestas.

  1. Cuales de entre los algoritmos de Dijkstra, Bellman-Ford, Johnson y Floyd-Wharshall son correctos cuando el grafo tiene pesos negativos?
  2. Considera un grafo no dirigido y conexo $G = (V, E)$ que tiene asignados pesos positivos en las aristas $(w_e)_{e \in E}$, y un subconjunto no vacío de vértices $U \subseteq V$ tal que $G \setminus U$ es conexo. Existe un árbol de expansión con coste mínimo T en el que todos los vértices de U son hojas?
  3. Dados una grafo dirigido y con capacidades positivas en sus aristas $G = (V, E, c_{e \in E})$ y dos vértices $s, t \in V$, una arista con capacidad mínima forma parte siempre de un s, t-corte con capacidad mínima?
  4. Qué condiciones tiene que cumplir una cadena de Markov para poder garantizar que existe una distribución estacionaria?

- **Ejercicio 2 (3 puntos)**
  Un palíndromo es una palabra tal que si la leemos del revés obtenemos la misma palabra, por ejemplo `ojo` es un palíndromo. Una cadena de caracteres $A = a_1 \cdots a_n$ admite una descomposición en palíndromos si, para algún valor k, podemos dividirla en k segmentos $0 = i_0 < i_1 < i_2 < i_k = n + 1$, de manera que, para cada $0 < j \leq k$, la subpalabra entre las posiciones i_{j-1} y i_j es un palíndromo.

  En particular siempre tenemos una descomposición en palíndromos trivial en la que cada letra es un palíndromo, pero la cadena puede admitir otras descomposiciones en palíndromos. Por ejemplo para la palabra $abbabbabbabba$ tenemos (entre otras) las descomposiciones en palíndromos $a|bb|a|bb|abba|bb|a$ y $abba|bb|abba|bb|a$

  Diseñad un algoritmo que dada una cadena A con n caracteres determine una descomposición en palíndromos con número mínimo de subpalabras.

- **Ejercicio 3 (3 puntos)**

  CoopC es una compañía de alquiler de coches que quiere diversificar su negocio para competir en los desplazamientos cortos. La compañía tiene oficinas distribuidas en Barcelona y un depósito central de coches en el Prat. La compañía quiere un método que le permita procesar las peticiones de trayectos cortos recibidos para el día siguiente. Para ello quiere determinar el número de coches que tiene que dejar en cada oficina al inicio del día dedicados a trayectos cortos. Esta asignación tiene que permitir tener coches disponibles suficientes a lo largo del día (en cada una de las oficinas) para poder cubrir todos los trayectos cortos del día. Además CoopC quiere minimizar el número de coches destinados durante el día a desplazamientos cortos.

  La información de la que dispone CoopC es la siguiente. Para cada solicitud de trayecto corto para el próximo día, dispone de una tupla (l, t, l', t') dónde el par (l, t) indica la oficina y la hora de recogida del vehículo y (l', t') indica la oficina y la hora de devolución.

  A efectos del esta aproximación CoopC asume que todos los coches son idénticos y que dispone de una flota suficientemente grande para cubrir cualquier nivel de demanda de trayectos cortos. Ademas, asume también que los coches serán recogidos y devueltos en las horas y locales estipulados.

  Se pide diseñar un algoritmo voraz que permita obtener el número de coches que se debe asignar a cada oficina, de manera que, a lo largo del día, siempre haya al menos un coche disponible en una oficina en la hora en la que algún cliente tiene que recogerlo allí. Teniendo en cuenta que mantener inmovilizado un coche tiene un coste alto la asignación obtenida debe garantizar el requisito de disponibilidad y asignar el menor número posible de coches a cada oficina.

- **Ejercicio 4 (3 puntos)**

  El servicio de control alimentario de la Generalitat está intentando acreditar un procedimiento de medida de la calidad alimentaria de productos complejos. En el laboratorio disponen m cromatógrafos y tienen técnicas de análisis de n componentes. Para cada cromatógrafo j se conoce conjunto S_i de componentes que puede analizar.

  Para acreditar el procedimiento necesitan garantizar que se han realizado como mínimo k análisis en cromatógrafos diferentes de cada una de las componentes. Dependiendo del nivel de independencia se requieren condiciones adicionales.

  - Nivel A: Un cromatógrafo solo puede utilizarse para analizar un máximo de dos componentes.
  - Nivel B: Los cromatógrafos del laboratorio son de tres marcas diferentes. En este nivel se requiere además que los cromatógrafos que analicen una componente no sean todos de la misma marca.

  Diseñad algoritmos para resolver los siguientes problemas:

  - Dados n, m, k y para cada cromatógrafo j, $1 \leq j \leq m$, los conjuntos $S_j \subseteq \{1, \ldots n\}$, determinar si es posible realizar análisis de las n componentes con el nivel de acreditación A.
  - Dados n, m, k y para cada cromatógrafo j, $1 \leq j \leq m$, los conjuntos $S_j \subseteq \{1, \ldots n\}$ y la marca $m_j \in \{1, 2, 3\}$, determinar si es posible realizar análisis de las componentes con el nivel de acreditación B.

  Los dos algoritmos, siempre que sea posible, deben obtener además una asignación de cromatógrafos a componentes verificando las condiciones requeridas por el nivel de acreditación.

Apellidos	Nombre	DNI

Problemas seleccionados

Ejercicio 1