1er Parcial: parte presencial

Algoritmos y Estructuras de Datos 3 – DC, FCEyN, UBA

14/10/2020

La parte presencial es una continuación de la parte domiciliaria. Por completitud, se incluye en este documento el enunciado de la parte domiciliaria pero en esta parte del examen sólo deben entregar la resolución de los nuevos incisos de **exactamente un ejercicio** a elección suya. Observar que estos incisos hacen referencia al enunciado original. De la misma forma, en la resolución de los ejercicios se puede hacer referencia a la entrega domiciliaria. En resumen, el parcial funciona como un todo.

El examen transcurre de 19:00 a 20:30 hs. Se supone que resuelvan el examen hasta las 20:00 hs. y, por tanto, de 20:00 a 20:30 no contestaremos más consultas de enunciado y sólo atenderemos cuestiones de entrega. Esa media hora está para que realicen la entrega sin problemas mediante el Campus (hasta las 20:30 hs. en `https://campus.exactas.uba.ar/mod/assign/view.php?id=142756`). Sólo en caso de que el Campus estuviera saturado y no funcionara, sería adecuado realizar la entrega por mail a algo3-doc@dc.uba.ar. Sólo en caso de entregar por Campus, el archivo podrá sobreescribirse una cantidad ilimitada de veces hasta la hora de entrega y, al igual que para el parcial domiciliario, deberán confirmar su entrega definitiva (que ya no podrá sobreescribirse).

El examen **debe** realizarse a mano. **Deben** numerar sus hojas y escribir en ellas sus nombres. Al finalizar, deben escanearlo o fotografiarlo. Nuevamente, el resultado debe ser un documento legible (buena iluminación, buena resolución, buena orientación, no fotos cortadas, etc.). Deben subir o bien un único archivo PDF o bien un único archivo comprimido que contenga las imágenes. En cualquier caso, **debe** estar nombrado `apellido_nombre.extensión` y **debe** haber un orden de lectura claro (pueden nombrar los archivos de las imágenes de acuerdo al orden de lectura).

Para realizar consultas, conectarse al aula de Zoom habitual (`https://zoom.us/my/dc.aula03`) y avisar que desean consultar. Serán agregades a una lista de espera y, cuando llegue su turno, transferides a una *breakout room* con algún docente al que realizarle su consulta.

Las aclaraciones de enunciado que podamos llegar a hacer van a ser comunicadas por Zoom y por Discord.

1) Decimos que un digrafo (con *loops*) tiene forma de ρ cuando todos sus vértices tienen grado de salida igual a 1.

   a) Dibujar 4 digrafos conexos y no isomorfos entre sí que tengan 6 vértices y forma de ρ.[1]

   b) Demostrar que si un digrafo es conexo y tiene forma de ρ entonces tiene un único ciclo.[2]

   c) Diseñar un algoritmo para encontrar un ciclo de longitud máxima en un digrafo con forma de ρ (no necesariamente conexo).

   Consideremos un período de tiempo circular $[0, T]$ tal que llegado el momento T se vuelve a contabilizar el tiempo 0 (e.g., un día, una semana, etc). Dentro del tiempo $[0, T]$ se encuentran definidas n actividades, la i-ésima de las cuales se desarrolla empezando en el instante s_i y terminando en el instante t_i (si $s_i > t_i$, entonces la actividad contiene el instante $0 = T$, ver Figura 1).

   En el problema de selección de actividades periódicas se busca determinar la máxima cantidad de actividades que un agente puede realizar rutinariamente, suponiendo que el agente es capaz de realizar una única actividad en cada instante. Formalmente, el objetivo es determinar la máxima razón x/y para una secuencia circular de actividades $A_1, \ldots, A_x$ que se realiza en y períodos completos cuando A_{i+1} se inicia lo antes posible una vez terminado A_i, para todo $1 \leq i \leq x$ (con $A_{x+1} = A_1$). Considere

[1] Decimos que un digrafo es conexo cuando su grafo subyacente es conexo.

[2] Si $v \to v$ es un *loop*, entonces v, v es un ciclo.

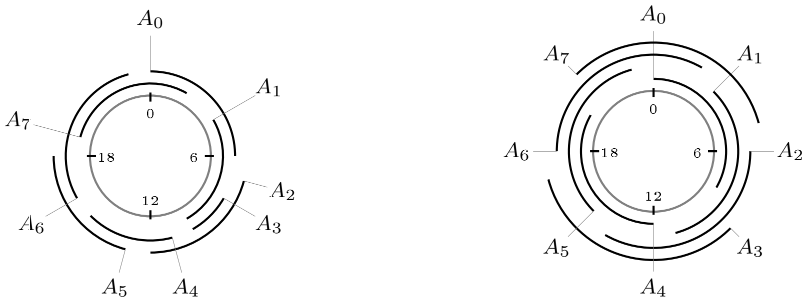

Figura 1: Ejemplos para horizontes de 24 horas con selecciones óptimas de actividades: izquierda = A_0, A_3, A_4, A_6 con razón 4; derecha = A_0, A_3, A_6, A_1, A_4, A_7, A_2, A_5 con razón 8/3.

la siguiente estrategia golosa para decidir qué actividad j conviene elegir si se elige la actividad i: tomar j como una actividad que no se solapa con i y cuyo tiempo de finalización es el primero desde t_i en un recorrido del tiempo en el sentido de las agujas del reloj.

Definir el digrafo de actividades D que tiene un vértice i por cada actividad y que tiene un arco (arista) $i \to j$ cuando j es la elección golosa que se toma si se elige i.

d) Demostrar que D es un digrafo con forma de ρ.

e) opcional) Demostrar que el ciclo máximo de D es una solución al problema de selección de actividades. **Ayuda:** demostrar por inducción en i ($i \leq x$) que existe una solución $A_1, \ldots, A_x$ donde A_{i+1} es la elección golosa para A_i, tomando $A_{x+1} = A_1$.

f) Usando los resultados anteriores, dar un algoritmo para resolver el problema de selección de actividades, suponiendo que las actividades se describen usando un conjunto de pares $[s_i, t_i]$ donde $0 \leq s_i, t_i \leq T$, $s_i \neq t_i$ y $T = 2n$.

Parte presencial. Sea $k = \lfloor x/y \rfloor$ para una solución óptima golosa $A_1, \ldots, A_x$ del problema de selección de actividades periódicas que tiene razon x/y.

g) Demostrar que el agente no puede realizar $k+1$ actividades en un período (considerando cualquier conjunto de actividades, no solo las de la solución óptima).

h) Demostrar que el agente puede realizar las actividades $A_i, \ldots, A_{i+k-1}$ en un período, para todo $1 \leq i \leq x$, donde $A_{x+i} = A_i$ para todo i.

i) Dar un algoritmo para encontrar un conjunto de actividades de cardinalidad máxima que puede realizar el agente en un período.

2) Se tiene una hilera con n recipientes que tienen $r_1, \ldots, r_n$ unidades de agua ($r_i \in \mathbb{N}$, $r_i < n$). Se quiere traspasar el agua de los recipientes a dos baldes con capacidades B_1 y B_2 ($B_1, B_2 \in \mathbb{N}$, $B_1, B_2 < n$). Para ello, en cada paso se toma el primer recipiente de la hilera y se vuelca completamente en uno de los baldes. Esta operación se repite mientras algún balde tenga una capacidad suficiente para albergar toda el agua del recipiente. El objetivo es maximizar la cantidad de recipientes que se vuelcan.

a) Escribir una función recursiva $f(i, b)$ para este problema que tenga la siguiente semántica: $f(i, b)$ es la máxima cantidad de recipientes que se pueden volcar suponiendo que ya se volcaron los recipientes $r_1, \ldots, r_{i-1}$ y el balde de capacidad B_1 tiene una capacidad remanente de b unidades de agua. **Ayuda:** observar que ya se volcaron $r_1 + \ldots + r_{i-1}$ unidades y, por lo tanto, se puede determinar la capacidad remanente del balde con capacidad B_2.

b) Mostrar que f tiene la propiedad de superposición de subproblemas para el caso considerado.

c) Diseñar un algoritmo de programación dinámica *top-down* para resolver el problema, indicando cuál es la invocación a la función que permite resolver el problema.

d) Determinar la complejidad temporal del algoritmo *top-down*.

Parte presencial.

e) Diseñar un algoritmo de programación dinámica *bottom-up* para resolver el problema, incluyendo pseudocódigo.

f) Comparar su algoritmo *bottom-up* con su algoritmo *top-down* de la parte domiciliaria.

5) Para organizar el tráfico, la ciudad de Ciclos Positivos ha decidido implementar las cabinas de peaje inverso. La idea de estas cabinas es incentivar la circulación de vehículos por caminos alternativos, estableciendo un monto que se le paga a le conductore de un vehículo cuando pasa por la cabina. Estas cabinas inversas se suman a las cabinas regulares, donde le conductore paga por pasar por la cabina. Dado que no tenemos empleo y aún no vendimos nuestro vehículo, queremos explotar un nuevo negocio que consiste en transitar por la ciudad de Ciclos Positivos a fin de obtener una ganancia que nos permita subsistir. Para ello, obtuvimos la información del costo c_i de transitar por cada cabina i de peaje ($c_i < 0$ si la cabina es inversa) y del costo c_{ij} que cuesta viajar de forma directa de cada cabina i a cada cabina j en caso de que esto sea posible (es decir, no pasando por otras cabinas intermedias).

a) Modelar como un problema de grafos el problema de determinar si es posible obtener una ganancia recorriendo eternamente las cabinas de la ciudad.

b) Dar un algoritmo para resolver el problema del inciso anterior, indicando su complejidad temporal.

Parte presencial. Desde la implementación del sistema de cabinas inversas, la administración de la ciudad de Ciclos Positivos observó que varies conductores se pasan cada día transitando continuamente sobre la red vehicular, agregando aun más congestión al sistema. A fin de remover los ciclos de pago positivo de la red sin abandonar la idea original ya promocionada, desde el departamento de *marketing* sugieren transformar algunas cabinas inversas en cabinas mixtas. Cuando un vehículo pasa por una cabina mixta, se le paga a le conductore si se le cobró a le conductore en la cabina anterior; caso contrario, le conductore paga. A fin de evitar malos resultados en el siguiente turno electoral, desde *marketing* sugieren que se le pague a le conductore cuando la cabina mixta sea la primera cabina recorrida.

c) Modelar como un problema de grafos el problema de determinar si es posible obtener una ganancia recorriendo eternamente las cabinas de la ciudad. Además de la información utilizada para el problema original, ahora se conoce cuáles cabinas son mixtas: notar que el monto de cobro es c_i y el monto de pago es $-c_i$ para la cabina mixta i (con $c_i > 0$). **Ayuda:** pensar cómo se pueden modelar las cabinas mixtas usando una cabina positiva y otra negativa. Luego, describir el grafo resultante.