Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC1253 - Matemáticas Discretas

Interrogación 2

11 de noviembre de 2020
Profesores: Gabriel Diéguez - Fernando Suárez

Instrucciones

- La duración de la interrogación es 1 hora y 20 minutos.
- Se podrán realizar preguntas solamente durante el módulo de clases, de 8:30 a 10:00 hrs., por el foro de Canvas correspondiente.
- Debe entregar una solución escrita a mano, ya sea en papel o tablet, antes de las 23:59 horas del día de la interrogación. En el caso de hacerlo en papel, debe preocuparse que la copia digital sea legible. Se recomienda el uso de algún software de escaneo como CamScanner o la app de Google Drive.
- Responda cada pregunta en archivo separado, `numalumno-P1.pdf` y `numalumno-P2.pdf`, y ponga su nombre y sección en cada archivo.
- En caso de hacer la interrogación fuera del horario, se recomienda tomar el tiempo (1 hora y 20 minutos) y entregarlo justo después de concluido el tiempo.
- Durante la evaluación puede hacer uso de sus apuntes o slides del curso.
- Esta es una evaluación estrictamente individual y, por lo tanto, no puede compartir información con sus compañeros o usar material fuera de sus apuntes o slides del curso. En caso de hacerlo, la interrogación no reflejará su progreso en el curso, viéndose perjudicada su formación personal y profesional.
- **Al comienzo de cada pregunta debe escribir la siguiente oración y firmarla:** *"Doy mi palabra que la siguiente solución de la pregunta X fue desarrollada y escrita individualmente por mi persona según el código de honor de la Universidad."*

  En caso de no escribir la oración o no firmarla, su solución no será evaluada.
- Entregue al menos una hoja por pregunta.
  - Si entrega la pregunta **completamente en blanco**, tiene nota mínima 1.5 en vez de 1.0 en la pregunta entregada.

Pregunta 1 [Grafos, Relaciones de Orden y Cardinalidad]

Sea $\mathcal{G}_i = \{G = (V, E) \mid |V| = i\}$, $i \in \mathbb{N}$

a) [2 pts] Demuestre que $(\mathcal{G}_i, \subseteq)$ es un orden parcial. ($G_1 \subseteq G_2$ si y solo si G_1 es un subgrafo de G_2)

b) [2 pts] Determine $\inf(\mathcal{G}_i)$ y $\sup(\mathcal{G}_i)$. Justifique sus respuestas.

c) [2 pts] Demuestre que $\mathcal{G} = \bigcup_{i=1}^{\infty} \mathcal{G}_i$ es numerable.

Solución

<u>**OBSERVACIÓN IMPORTANTE:**</u> El enunciado de esta pregunta no especificó el conjunto de donde se obtenían los posibles vértices de los grafos, por lo que tomando distintos supuestos las preguntas *b*) y *c*) pueden tener respuestas diferentes. En esta solución, se supondrá que

$$\mathcal{G}_i = \{G = (V, E) \mid V = \mathbb{N}_i\}$$

donde $\mathbb{N}_i$ es el conjunto de los naturales módulo i visto en clases.

a) Debemos demostrar que la relación "ser subgrafo" es una relación de orden parcial. Sean grafos $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ y $G_3 = (V_3, E_3)$.

   1) <u>Refleja:</u> Como $V_1 \subseteq V_1$ y $E_1 \subseteq E_1$, se cumple que $G_1 \subseteq G_1$.
   2) <u>Antisimétrica:</u> Supongamos que $G_1 \subseteq G_2$ y $G_2 \subseteq G_1$. Por definición de subgrafo esto significa que
   $$V_1 \subseteq V_2 \wedge V_2 \subseteq V_1 \rightarrow V_1 = V_2$$
   $$E_1 \subseteq E_2 \wedge E_2 \subseteq E_1 \rightarrow E_1 = E_2$$
   y por lo tanto $G_1 = G_2$. Se concluye que la relación es antisimétrica.
   3) <u>Transtiva:</u> Supongamos que $G_1 \subseteq G_2$ y $G_2 \subseteq G_3$. Por definición de subgrafo esto significa que
   $$V_1 \subseteq V_2 \wedge V_2 \subseteq V_3 \rightarrow V_1 \subseteq V_3$$
   $$E_1 \subseteq E_2 \wedge E_2 \subseteq E_3 \rightarrow E_1 \subseteq E_3$$
   y por lo tanto $G_1 \subseteq G_3$. Se concluye que la relación es transitiva.

b) Notemos que dado el supuesto de que todos los grafos en $\mathcal{G}_i$ tienen el mismo conjunto de vértices, lo único que diferenciará a estos grafos son sus aristas.

   Para el supremo, el mayor conjunto de aristas que podemos tomar es $V \times V$, por lo que el supremo será el grafo completo de i vértices K_i. Recordemos que por definición E es una relación binaria sobre V, y luego $E \subseteq V \times V$, por lo que cualquier otro grafo en $\mathcal{G}_i$ será subgrafo de K_i.

Para el ínfimo, es claro que el menor conjunto de aristas posibles es $\varnothing$, y por lo tanto el ínfimo es un conjunto independiente de tamaño i.

c) Dado nuestro supuesto, cada $\mathcal{G}_i$ tiene una cantidad finita de elementos, pues la cantidad de relaciones binarias posibles sobre un conjunto finito es finita. Luego, podemos enumerar los elementos de $\bigcup_{i=1}^{\infty} \mathcal{G}_i$ fácilmente, enumerando primero todos los posibles grafos en $\mathcal{G}_0$, luego en $\mathcal{G}_1$ y así sucesivamente. Como en cada caso son finitos, si esperamos lo suficiente aparecerá cualquier grafo en $\bigcup_{i=1}^{\infty} \mathcal{G}_i$. Se concluye entonces que el conjunto es enumerable.

Pauta (6 pts.)

a)
- 0.5 ptos por demostrar que la relación es refleja.
- 0.75 ptos por demostrar que la relación es antisimétrica.
- 0.75 ptos por demostrar que la relación es transitiva.

b)
- 1 pto por dar y justificar el ínfimo.
- 1 pto por dar y justificar el supremo.

c) 2 ptos.

Puntajes intermedios y soluciones alternativas a criterio del corrector.

Pregunta 2 [Algoritmos y Complejidad]

Definimos un *cerro* como una lista consistiendo en una serie estrictamente creciente seguida de una serie estrictamente decreciente. Por ejemplo:

$$[1, 2, 4, 19, 8, 3]$$

$$[-5, 2, 7, 10, 15, 6, 5, 4, 1, 0]$$

son *cerros*.

a) [3 pts] Escriba un algoritmo que utilice una estrategia de dividir y conquistar que reciba como input un *cerro*, y entregue como output el valor máximo de este.

b) [3 pts] Calcule la complejidad de su algoritmo.

Para obtener todo el puntaje en ambas preguntas su algoritmo debe ser $O(log(n))$.

Solución

a) **Input:** un cerro $A = [a_0, \dots, a_{n-1}]$ de largo n.
Output: el valor máximo de A.

CerroSearch($A = [a_0, \dots, a_{n-1}]$)
$a = 0$
$b = n - 1$
$m = \lfloor \frac{a+b}{2} \rfloor$
if $b == 0$ **then**
  return $A[0]$
else if $A[m] > A[m+1]$ **then**
  return CerroSearch($A[: m+1]$)
else if $A[m] < A[m+1]$ **then**
  return CerroSearch($A[m+1 :]$)
end if

b) Buscamos determinar la cantidad de comparaciones $T(n)$, donde n es el largo del input. Consideremos la siguente ecuación de recurrencia:

$$T(n) = \begin{cases} 1 & n = 1 \\ 3 + T(\lceil n/2 \rceil) & n > 1 \end{cases}$$

Utilizaremos el reemplazo $n = 2^k$ con $k \in \mathbb{N}$.

Luego, $T(2^k) = 3 + T(2^{k-1})$, siguiendo esta ecuación de recurrencia resulta:

$$
\begin{aligned}
T(n) &= 3 + 3 + T(2^{k-2}) \\
T(n) &= \sum_{i=1}^{2} 3 + T(2^{k-2}) \\
&\vdots \\
T(n) &= \sum_{i=1}^{k} 3 + T(2^{k-k}) \\
T(n) &= \sum_{i=1}^{k} 3 + 1 \\
T(n) &= 3k + 1
\end{aligned}
$$

donde $k = \log_2(n)$, luego podemos reescribir la recurrencia como:

$$T(n) = 3\log_2(n) + 1$$

con lo que

$$T(n) \in O(\log_2(n) \mid \text{POTENCIA}_2)$$

Además, dado que $\log_2(n)$ es asintóticamente no decreciente, 2-armónica y $T(n)$ es asintóticamente no decreciente[1] concluimos que:

$$T(n) \in O(\log_2(n))$$

Pauta (6 pts.)

a) 3.0 ptos.

b) 3.0 ptos.

Puntajes intermedios y soluciones alternativas a criterio del corrector.

[1] Visto en clases.