Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2413 — Bases de Datos — Sección 3 — 2'2026

PAUTA INTERROGACIÓN 2

Preguntas en blanco: Items entregados en blanco se evaluarán con 0.25 de 6 puntos.

Pregunta 1

Considere la siguiente entrevista entre usted y un mago que desea crear una base de datos para almacenar la información de su Asociación de Magos.

Usted *Hola, Gandalf. Gracias por recibirme. He estado analizando los requerimientos que me enviaste para el sistema de la Asociación de Magos. Empecemos por ustedes: los Magos. ¿Qué datos mínimos necesitan registrar de cada miembro?*

Gandalf ¡Ah, bienvenido, amigo mío! Verás, un mago es un misterio, pero para el gremio necesitamos su nombre... el verdadero no, por supuesto, sino su nombre artístico, único para cada mago de la asociación. También el año en que recibieron su primer sombrero de copa, que es cuando se vuelven profesionales. Y, claro está, su gran poder.

Usted *Cuando dices "su gran poder", ¿te refieres a una especialidad técnica? ¿Cómo cartomagia, escapismo o mentalismo?*

Gandalf Sí, algo así, aunque algunos dominan varias artes. Merlín, por ejemplo, te hace una desaparición de elefantes en el escenario y luego se sienta a leerte la mente con las cartas. Es un caos etiquetarlo, pero siempre logramos darle una única especialidad. Pero lo verdaderamente importante son sus trucos. Cada mago tiene su repertorio.

Usted *Mencionas que Merlín hace varios trucos. ¿Esos trucos son de su propiedad exclusiva o cualquier otro mago de la asociación puede aprenderlos y realizarlos?*

Gandalf ¡Por los dioses, la magia es libre! Cada truco tiene su tiempo y dificultad que puede ser baja, media o alta. Un truco clásico como "La mujer cortada en dos" lo hacen al menos veinte de nuestros magos. Claro que la versión de Merlín dura quince minutos porque le mete mucho drama, mientras que el joven Houdini lo hace en cinco minutos porque va directo a la acción.

Usted *(Sonriendo) Entendido. La duración no depende del truco en sí, sino de quién lo ejecuta. Ahora hablemos de las herramientas: los Aparatos de magia. ¿Cómo los controlan?*

Gandalf Las herramientas... nuestro dolor de cabeza. Tenemos un gran almacén central. Ahí guardamos desde las sábanas de seda hasta las grandes cajas de desaparición y las guitarras levitadoras. Cada objeto físico es único, tiene un código grabado en la madera o el metal. El problema es que a veces un mago viene, se lleva una caja para un truco y la devuelve rota. Por lo que, cada vez que un aparato sufra un desgaste o reparación, registramos la fecha y el arreglo, ligado siempre a ese aparato. El mes pasado la "Guillotina del Terror" falló dos veces, solo aguanto unos días antes de fallar de nuevo. Necesito saber cuántas veces ha ido al taller antes de que ocurra una desgracia en el escenario.

Usted *Queda anotado. Ahora, volviendo a los trucos... ¿Cada aparato sirve para un solo truco específico, o usan el mismo aparato para trucos distintos?*

Gandalf Qué pregunta tan capciosa. A ver, la "Caja de Doble Fondo #04" se construyó exclusivamente para el truco de "La Desaparición del Conejo". No se puede usar para otro truco o podría suceder un accidente. Asi que no, cada aparato tiene su único truco asociado, aunque un truco puede usar muchos aparatos o ninguno.

Usted *Interesante. Lo que me lleva a mi última pregunta. ¿cuando usan esos trucos?*

Gandalf ¡Los shows! La razón de ser de la Asociación. Nos contratan en teatros, hoteles o eventos corporativos. Cada show tiene un nombre único, una fecha y un lugar. En un show pueden actuar varios magos y cada mago puede ocupar uno o más aparatos. Toda esa información nos interesa dejarla bien registrada.

Usted *Muy bien, ya tenemos todo lo necesario para diseñar su base de datos.*

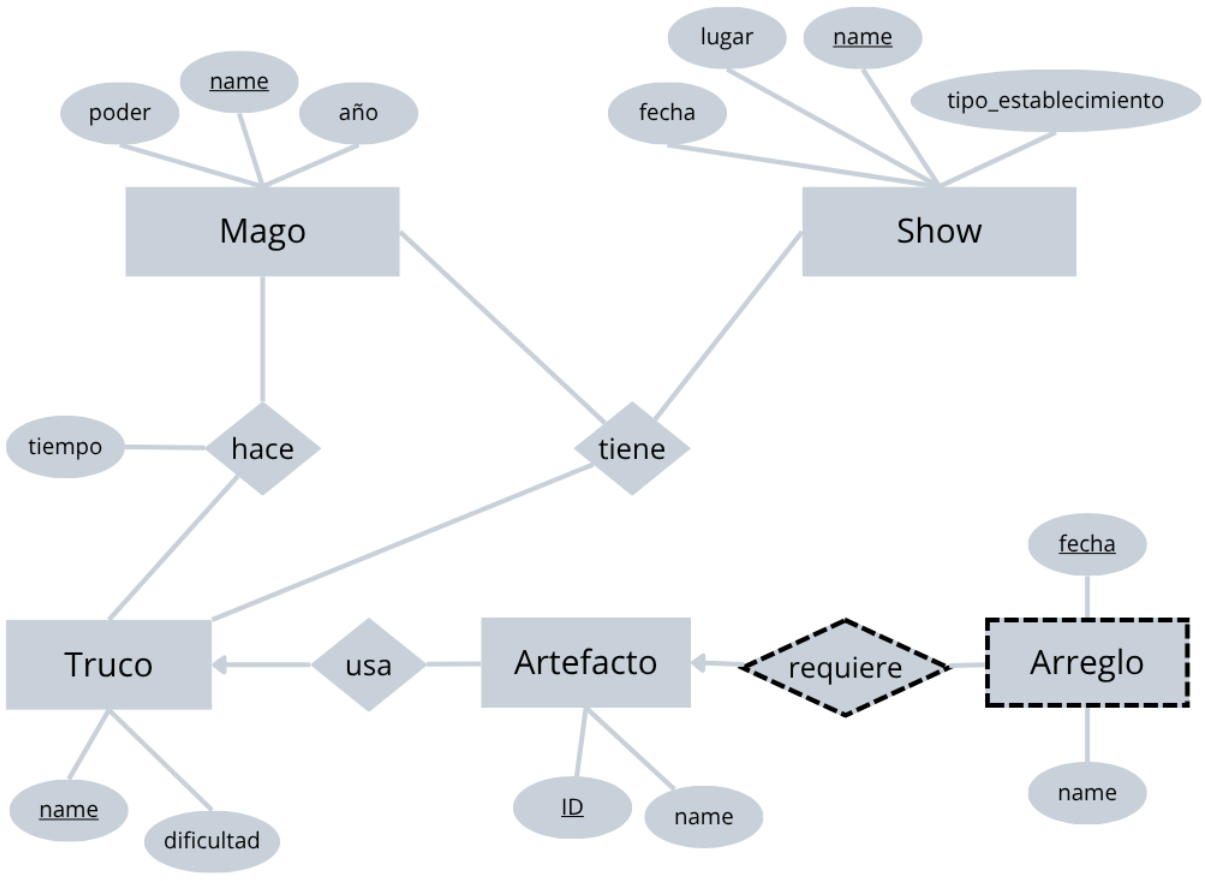
Gandalf (Abriendo los ojos con asombro) ¡Por las barbas de Merlín! Eso no es una base de datos, ¡eso es auténtica alta hechicería!

1. Dada la entrevista anterior, dibuje un diagrama entidad/relación para el dominio identificando las entidades, atributos, llaves, relaciones y tipo de cada relación. En caso de alguna información sea ambigua, defina sus supuestos y argumentelos. Por último, explique su diagrama y decisiones.

2. Convierta su diagrama anterior a un esquema relacional especificando todas las relaciones, sus nombres de atributo, llaves primarias subrayadas, llaves foráneas y a qué relación apuntan.

Solución 1.1

Respuesta tentativa:

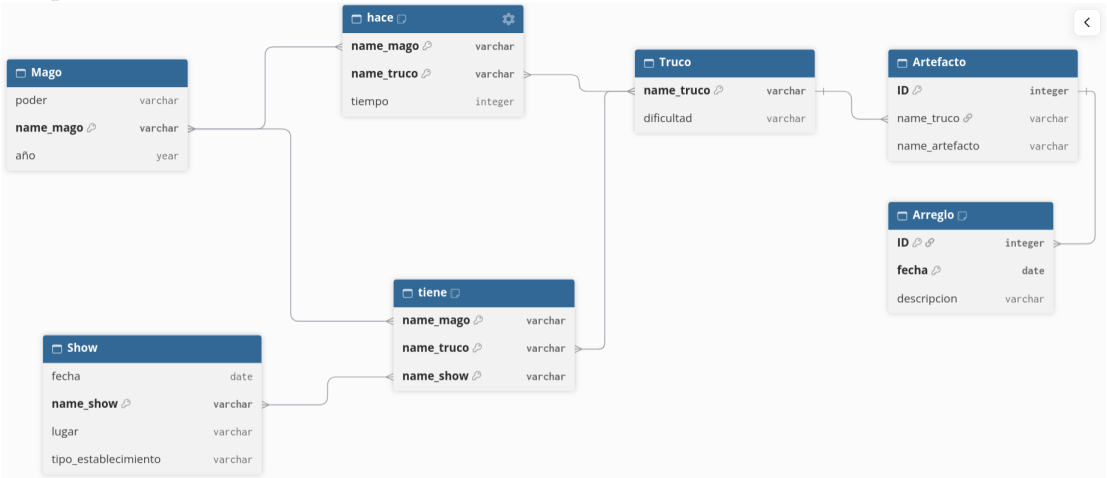


Puntaje

- **(0.5 puntos)** Entidades (5): Mago, Show, Truco, Artefacto y Arreglo.
  - Se otorga el puntaje completo si aparecen las cinco, Arreglo debe ser entidad y no atributo, pues tiene fecha y descripción propias.
- **(0.5 puntos)** Relación "hace" + atributo tiempo.
  - Relación muchos a muchos entre Mago y Truco (un truco lo hacen muchos magos y un mago hace muchos trucos).
  - La duración depende de quién ejecuta el truco, por lo que tiempo es atributo de la relación y no de Truco.
- **(0.5 puntos)** Relación "usa" de 1 a n.
  - Cada Artefacto sirve a un único Truco (flecha hacia Truco), un Truco puede usar muchos artefactos o ninguno.
- **(0.5 puntos)** Relación "requiere" de 1 a n y débil.
  - Un Artefacto tiene muchos Arreglos, pero cada Arreglo pertenece a un solo Artefacto.
  - Arreglo es entidad débil, no se identifica sin su Artefacto, su llave parcial es fecha.
- **(0.5 puntos)** Relación "tiene" ternaria.
  - Une Mago, Show y Truco, registra qué mago realiza qué truco en qué show.
  - Debe ser ternaria, ya que dos relaciones binarias separadas pierden la información de la combinación.
- **(0.5 puntos)** Atributos (mínimos) y llaves pertinentes a la entidad.
  - Mago: nombre (artístico, único), año, poder.
  - Show: nombre, fecha, lugar.
  - Truco: nombre, dificultad.
  - Artefacto: ID (código grabado), nombre.
  - Arreglo: fecha (llave parcial), descripción.

Solución 1.2

Respuesta tentativa:



Puntaje

- **(0.5 puntos)** Tablas (7): Mago, Show, Truco, Artefacto, Arreglo, hace y tiene.
  - "usa" no genera tabla propia, ya que se absorbe en Artefacto. Por su lado, "requiere" no puede ser tabla ya que Arreglo necesita el ID del Artefacto para poder tener su llave.
- **(0.5 puntos)** Tabla "hace" con atributo *tiempo*.
  - hace(name_mago, name_truco, tiempo), con *name_mago* y *name_truco* como llaves foráneas.
- **(0.5 puntos)** Tabla "tiene" ternaria.
  - tiene(name_mago, name_show, name_truco): llave primaria compuesta por los tres atributos, cada uno foránea a su tabla.
- **(0.5 puntos)** Llaves primarias correctas (subrayadas) en cada tabla.
  - Simples: Mago(name_mago), Show(name_show), Truco(name_truco), Artefacto(ID).
  - Compuestas: Arreglo(artefacto, fecha), hace(name_mago, name_truco) y tiene (los tres atributos).
  - Arreglo debe incluir la llave del Artefacto en su llave primaria por ser entidad débil.
- **(0.5 puntos)** Llaves foráneas indicadas, con la tabla a la que apuntan.
  - Artefacto.name_truco → Truco.name_truco (viene de "usa").
  - Arreglo.artefacto → Artefacto.ID (viene de "requiere").
  - hace.name_mago → Mago, hace.name_truco → Truco.
  - tiene.name_mago → Mago, tiene.name_show → Show, tiene.name_truco → Truco.
- **(0.5 puntos)** Atributos pertinentes en cada tabla.
  - Mago(name_mago, año, poder); Show(name_show, fecha, lugar, tipo_establecimiento); Truco(name_truco, dificultad); Artefacto(ID, nombre, truco, name_truco); Arreglo(artefacto, fecha, nombre).

Pregunta 2

1. Considere una relación $R(A, B, C, D, E, F, X, Y, Z)$ con dependencias funcionales:

$$\mathcal{F} = \{AB \to CD, CE \to F, AB \to E, X \to YZ, Z \to X\}$$

   Convierta R a BCNF explicando cada uno de sus pasos.

2. Sea $R(A_1, \ldots, A_n)$ una relación y $\mathcal{F}$ su conjunto de dependencias funcionales tal que $\mathcal{F} \neq \emptyset$. Demuestre que si existe i tal que A_i no aparece mencionado en $\mathcal{F}$, entonces R no esta en BCNF.

Solución 2.1

Nota: el desarrollo a continuación es un ejemplo considerando uno de los caminos posibles para abordar la pregunta (escogiendo la cuarta dependencia como base). El puntaje se define en base a logros generales y no al apego exacto a las decisiones expuestas a continuación. Cualquier descomposición correcta en BCNF obtenida aplicando el algoritmo visto en clases será considerada válida, asignando el puntaje de cada nivel de forma equivalente a los pasos realizados.

Primer Nivel

Escogemos la dependencia $X \to YZ$ y calculamos su clausura usando el algoritmo visto en clases:

$$X^+ = \{X, Y, Z\}$$ **(0.25 puntos)**

Como $X^+ \neq \text{att}(R)$, se tiene que X **no es superllave** de R, por lo que R no está en BCNF. **(0.25 puntos)**

Definimos $X^- = (\text{att}(R) \setminus X^+) \cup \{X\} = \{X, A, B, C, D, E, F\}$ y descomponemos R en $R_1 = \pi_{X^-}(R)$ y $R_2 = \pi_{X^+}(R)$:

$$R_1(X, A, B, C, D, E, F) \qquad R_2(X, Y, Z)$$ **(0.25 puntos)**

Calculando las dependencias de cada relación según $\mathcal{F}$ (Algoritmo 2):

$$\mathcal{F}_1 = \{AB \to CD,\ CE \to F,\ AB \to E\}$$
$$\mathcal{F}_2 = \{X \to YZ,\ Z \to X\}$$

En R_2 se cumple que $X^+ = Z^+ = \{X, Y, Z\} = \text{att}(R_2)$, por lo que el lado izquierdo de toda dependencia no trivial es superllave. Luego, R_2 **está en BCNF**, mientras que R_1 requiere seguir descomponiéndose. **(0.25 puntos)**

Segundo Nivel

Trabajamos con $R_1(X, A, B, C, D, E, F)$ y escogemos la dependencia $CE \to F$:

$$(CE)^+ = \{C, E, F\}$$ **(0.25 puntos)**

Como $(CE)^+ \neq \text{att}(R_1)$, CE **no es superllave** de R_1, por lo que no está en BCNF. **(0.25 puntos)**

Definimos $(CE)^- = (\text{att}(R_1) \setminus (CE)^+) \cup \{C, E\} = \{C, E, A, B, D, X\}$ y descomponemos R_1 en:

$$R_3(X, A, B, C, D, E) \qquad R_4(C, E, F)$$ **(0.25 puntos)**

con dependencias:

$$\mathcal{F}_3 = \{AB \to CD,\ AB \to E\}$$
$$\mathcal{F}_4 = \{CE \to F\}$$

En R_4 se tiene $(CE)^+ = \{C, E, F\} = \text{att}(R_4)$, por lo que CE es superllave y R_4 **está en BCNF**, mientras que R_3 requiere seguir descomponiéndose. **(0.25 puntos)**

Tercer Nivel

Trabajamos con $R_3(X, A, B, C, D, E)$ y escogemos la dependencia $AB \to CD$:

$$(AB)^+ = \{A, B, C, D, E\}$$ **(0.25 puntos)**

Como $X \notin (AB)^+$, se tiene $(AB)^+ \neq \text{att}(R_3)$, por lo que AB **no es superllave** de R_3. **(0.25 puntos)**

Definimos $(AB)^- = (\text{att}(R_3) \setminus (AB)^+) \cup \{A, B\} = \{A, B, X\}$ y descomponemos R_3 en:

$$R_5(A, B, X) \qquad R_6(A, B, C, D, E)$$ **(0.25 puntos)**

con dependencias:

$$\mathcal{F}_5 = \varnothing$$
$$\mathcal{F}_6 = \{AB \to CD,\ AB \to E\}$$

R_5 no tiene dependencias no triviales, por lo que está en BCNF. En R_6 se tiene $(AB)^+ = \{A, B, C, D, E\} = \text{att}(R_6)$, por lo que AB es superllave y R_6 también está en BCNF. Luego, **ambas relaciones están en BCNF**. **(0.25 puntos)**

Resultado final

La descomposición de R en BCNF es:

$$(R_2(X,Y,Z),\ \mathcal{F}_2) \qquad (R_4(C,E,F),\ \mathcal{F}_4)$$
$$(R_5(A,B,X),\ \mathcal{F}_5) \qquad (R_6(A,B,C,D,E),\ \mathcal{F}_6)$$

Puntaje

Dado lo anterior, la distribución de puntaje es la siguiente:
Primer nivel

- **(0.25 puntos)** Escoger una dependencia y definir su clausura.
- **(0.25 puntos)** Identificar que no es superllave.
- **(0.25 puntos)** Descomponer la relación en 2 relaciones equivalentes con dependencias utilizando X^+ y X^-.
- **(0.25 puntos)** Definir las dependencias funcionales de cada relación e identificar una que ahora satisface BCNF y otra que requiere seguir descomponiendo.

Segundo nivel

- **(1 punto)** Desglose idéntico al primer nivel con la relación que requería mayor descomposición. Nuevamente debiese quedar una relación en BCNF y una que requiere seguir descomponiendo.

Tercer nivel

- **(0.25 puntos)** Escoger una dependencia y definir su clausura.
- **(0.25 puntos)** Identificar que no es superllave.
- **(0.25 puntos)** Descomponer la relación en 2 relaciones equivalentes con dependencias utilizando X^+ y X^-.
- **(0.25 puntos)** Definir las dependencias funcionales de cada relación e identificar que ahora ambas satisfacen BCNF.

Solución 2.2

Sea $R(A_1, \ldots, A_n)$ con $\mathcal{F} \neq \varnothing$ (conjunto de dependencias funcionales no triviales), y supongamos que existe i tal que A_i no aparece mencionado en $\mathcal{F}$. Supongamos, por contradicción, que R **sí está** en BCNF.
Como $\mathcal{F} \neq \varnothing$, sea $\bar{X} \to \bar{Y} \in \mathcal{F}$ una dependencia funcional no trivial. Como R está en BCNF, $\bar{X}$ debe ser superllave de R, es decir:

$$\bar{X}^+ = \text{att}(R) \qquad \textbf{(1.0 punto)}$$

Por otro lado, como A_i no aparece en $\mathcal{F}$, entonces $A_i \notin \bar{X}^+$. Considerando el algoritmo visto en clases para calcular $\bar{X}^+$:

- La clausura se inicializa como $\{\bar{X}\}$, y $A_i \notin \{\bar{X}\}$, pues $\bar{X}$ es el lado izquierdo de una dependencia de $\mathcal{F}$ y A_i no aparece en $\mathcal{F}$.
- En cada iteración, la clausura solo se extiende agregando los atributos $\{\bar{C}\}$ del lado derecho de alguna dependencia $\bar{B} \to \bar{C} \in \mathcal{F}$. Como A_i no aparece en ninguna dependencia de $\mathcal{F}$, nunca se agrega a la clausura.

Por lo tanto:

$$A_i \notin \bar{X}^+ \qquad \textbf{(1.0 punto)}$$

Como $A_i \in \text{att}(R)$ pero $A_i \notin \bar{X}^+$, se tiene que $\bar{X}^+ \neq \text{att}(R)$, y por lo tanto $\bar{X}$ **no es superllave** de R. Esto contradice que R esté en BCNF, pues $\bar{X} \to \bar{Y}$ es una dependencia no trivial cuyo lado izquierdo no es superllave. Concluimos que R **no está en BCNF**. ■ **(1.0 punto)**

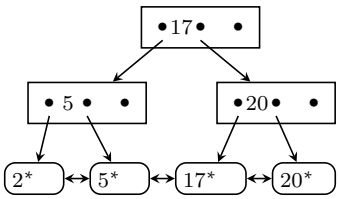
Puntaje

Dado lo anterior, la distribución de puntaje es la siguiente:

- **(1.0 punto)** Implementar demostración por contradicción y utilizar una dependencia perteneciente a $\mathcal{F}$.
- **(1.0 punto)** Razonar que como A_i no aparece en $\mathcal{F}$, entonces A_i no pertenece a la clausura del/los atributos a la izquierda de dependencia escogida. No argumentar el razonamiento implica un descuento de **-0.5 puntos**.
- **(1.0 punto)** Concluir que como A_i no pertenece a la clausura del/los atributos a la izquierda de dependencia escogida, entonces estos no son superllave y por ende la relación no está en BCNF.

Nota: si el estudiante explica correctamente pero no demuestra de manera formal se asignan 2/3 puntos.

Pregunta 3



1. Considere el B+tree de orden $d = 1$ de la figura. Entregue una secuencia de inserciones (sin redistribución) de largo mínimo que aumente la altura de este B+tree en uno. Explique su respuesta.
2. Sea un B+tree B de orden $d = 1$ y altura H, tal que cada página hoja tiene exactamente un dato y cada página interior tiene exactamente una entrada. La figura anterior es un ejemplo de este caso donde $H = 2$. Entregue una secuencia de inserciones (sin redistribución) de largo mínimo que aumente la altura de B en uno. Explique su respuesta.

Solución 3.1

Respuesta. Una secuencia de largo mínimo es

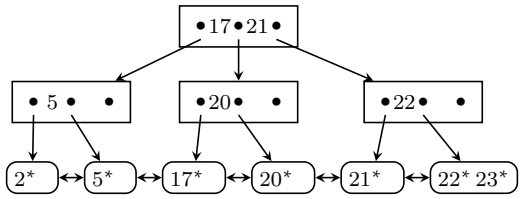
$$\boxed{21,\ 22,\ 23,\ 24,\ 25}$$

(5 inserciones). En general sirve cualquier secuencia *estrictamente creciente* de 5 valores mayores que 20, por ejemplo $30, 40, 50, 60, 70$.

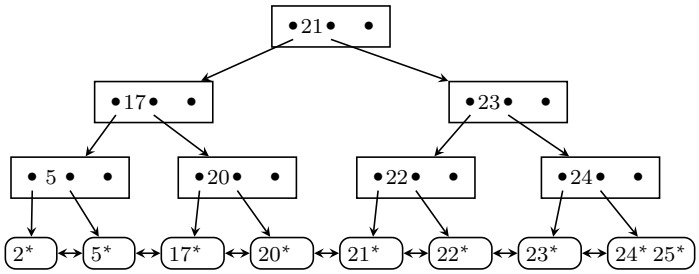
Traza de las inserciones

Inserción	Qué ocurre	Split hoja	Split directorio	Altura
21	Llega a la hoja $[20^*]$, que queda $[20^*\ 21^*]$.	No	No	2
22	Hoja $[20^*\ 21^*\ 22^*]$ desborda $\Rightarrow [20^*]\,\vert\vert\,[21^*\ 22^*]$, se copia 21 al padre: $[20] \to [20, 21]$.	Sí	No	2
23	Hoja $[21^*\ 22^*\ 23^*]$ desborda $\Rightarrow [21^*]\,\vert\vert\,[22^*\ 23^*]$, se copia 22: $[20, 21, 22]$ desborda $\Rightarrow [20]\,\vert\vert\,[22]$, sube 21 a la raíz: $[17] \to [17, 21]$.	Sí	Sí (1)	2
24	Hoja $[22^*\ 23^*\ 24^*]$ desborda $\Rightarrow [22^*]\,\vert\vert\,[23^*\ 24^*]$, se copia 23: $[22] \to [22, 23]$.	Sí	No	2
25	Hoja $[23^*\ 24^*\ 25^*]$ desborda $\Rightarrow [23^*]\,\vert\vert\,[24^*\ 25^*]$, se copia 24: $[22, 23, 24]$ desborda $\Rightarrow [22]\,\vert\vert\,[24]$, sube 23: raíz $[17, 21, 23]$ desborda $\Rightarrow [17]\,\vert\vert\,[23]$, sube 21 como **nueva raíz**.	Sí	Sí (2)	**3**

Después de insertar 23:



Árbol final (después de insertar 25), altura 3:



Explicación (por qué 5 es el mínimo)

Splits de directorio. La altura sólo crece si se divide la raíz [17]. Como tiene 1 clave, necesita recibir 2 claves más (llegar a 3), y cada clave que recibe viene de un split de un nodo del nivel intermedio. Por lo tanto se necesitan **al menos 2 splits de directorio** en el nivel intermedio. Cada nodo interior parte con 1 clave, y después de un split ambas mitades quedan nuevamente con 1 clave; así, cada split de directorio necesita que a ese nodo le lleguen 2 claves nuevas, es decir, **2 splits de hoja** bajo él. En total: $2 \cdot 2 = 4$ splits de hoja.

Splits de hoja. Todas las hojas parten con exactamente 1 dato, así que el primer split de hoja requiere 2 inserciones en la misma hoja. Además, una inserción produce a lo más un split de hoja. Luego, para 4 splits de hoja se necesitan al menos $4 + 1 = 5$ inserciones. La secuencia $21, 22, 23, 24, 25$ logra exactamente eso: como los valores son crecientes, cada inserción cae en la hoja de más a la derecha, que después de cada split queda con 2 datos, por lo que *toda* inserción desde la segunda provoca un split de hoja, y todos esos splits caen bajo el mismo camino hacia la raíz.

Puntaje

Dado lo anterior, la distribución de puntaje es la siguiente:

- **(2 puntos)** Secuencia:
  - **(1 punto)** La secuencia hace que *siempre* haya split de hoja (cada inserción desde la segunda divide una hoja; no hay inserciones "desperdiciadas").
  - **(1 punto)** La secuencia hace que *siempre* haya split de directorio cuando corresponde (los splits de hoja se concentran bajo el mismo camino y se propagan hasta dividir la raíz).
- **(1 punto)** Explicación:
  - **(0.5 puntos)** Explica correctamente los splits de hoja (hojas con 1 dato ⇒ el primer split cuesta 2 inserciones; luego cada inserción causa un split).
  - **(0.5 puntos)** Explica correctamente los splits de directorio (la raíz necesita 2 claves más ⇒ 2 splits intermedios ⇒ 4 splits de hoja).

Notas para corregir:

- Si el alumno usa la convención contraria de split de hoja (2 datos a la izquierda y 1 a la derecha) de forma consistente, la solución simétrica es decreciente, por ejemplo $16, 15, 14, 13, 12$ (todas caen en la hoja de 5*). Se acepta con puntaje completo.

Solución 3.2

Respuesta. Sea M el valor máximo almacenado en B. Una secuencia de largo mínimo es

$$\boxed{M+1,\ M+2,\ \ldots,\ M+K, \qquad K = 2^H + 1.}$$

H	Splits de hoja necesarios (2^H)	Inserciones $K = 2^H + 1$
0	1	$1 + 1 = 2$
1	2	$1 + 2 = 3$
2	4	$1 + 4 = 5$ (coincide con 3.1)
3	8	$1 + 8 = 9$

Explicación

Splits de directorio. Sea $S(h)$ el número de splits de hoja necesarios para que un nodo a altura h se divida (las hojas tienen altura 0, y $S(0) = 1$ porque la hoja misma se divide). Un nodo interior con 1 clave necesita recibir 2 claves para llegar a 3, y cada clave viene de un split de un hijo a altura $h-1$. Después de un split, el nodo nuevo vuelve a tener 1 clave, así que el segundo split del hijo cuesta lo mismo que el primero:

$$S(h) = S(h-1) + S(h-1) = 2\,S(h-1) \quad \Longrightarrow \quad S(H) = 2^H.$$

Equivalentemente: $S(H) = S(H-1) + S(H-2) + \cdots + S(1) + S(0) + 1$, y como $S(H-2) + \cdots + S(0) + 1 = S(H-1)$, se obtiene $S(H) = 2\,S(H-1) = 2^H$.

Splits de hoja. Todas las hojas parten con 1 dato, así que el primer split de hoja necesita 2 inserciones, y cada inserción produce a lo más un split de hoja. Por lo tanto

$$K \ \geq\ S(H) + 1 \ =\ 2^H + 1.$$

La secuencia $M+1, \ldots, M+2^H+1$ alcanza la cota: como cada valor es mayor que todos los anteriores, siempre cae en la hoja de más a la derecha. La primera inserción la deja con 2 datos; desde ahí, cada inserción divide esa hoja (que vuelve a quedar con 2 datos a la derecha) y manda una clave al camino derecho. Los nodos de ese camino funcionan como un *contador binario*: un nodo con 1 clave es un "0" y uno con 2 claves es un "1"; cada split de hoja suma 1 y los splits de directorio son los "acarreos". Con H niveles interiores el contador se desborda (se divide la raíz) exactamente después de 2^H splits de hoja.

Puntaje

Dado lo anterior, la distribución de puntaje es la siguiente:

- **(1 punto)** Secuencia: $M+1, M+2, \ldots, M+K$ (o cualquier secuencia estrictamente creciente de valores mayores que el máximo del árbol).
- **(1 punto)** Valor de K: $K = 2^H + 1$.
- **(1 punto)** Explicación:
  - **(0.5 puntos)** Explica los splits de hoja (el primer split cuesta 2 inserciones y luego cada inserción provoca un split, de ahí el "+1").
  - **(0.5 puntos)** Explica los splits de directorio (cada nivel necesita el doble de splits que el nivel de arriba, de ahí el 2^H; vía recurrencia $S(H)$, inducción o argumento equivalente).

Notas para corregir:

- Igual que en 3.1, se acepta la solución simétrica decreciente ($m-1, m-2, \ldots$, con m el mínimo) si el alumno usa de forma consistente la convención de split de hoja con 2 datos a la izquierda.
- Si el alumno mide la altura contando niveles en vez de aristas, la respuesta no cambia, siempre que sea consistente con que en la figura $H = 2$.