



BEBRAS PERÚ



Los retos del castor



#3





En este documento encontrarás desafíos con diferentes niveles de dificultad en los que se ponen en juego las habilidades del Pensamiento Computacional.

En cada desafío se plantean los problemas a resolver. Al final de cada problema encontrarás la respuesta y la explicación de la relación del problema con el Pensamiento Computacional.

Habilidades del Pensamiento Computacional

En el Pensamiento Computacional se ponen en juego una serie de estrategias cognitivas aplicadas a la resolución de problemas:



ABSTRACCIÓN:

centrar la atención solo en la información importante, ignorando los detalles irrelevantes.



DESCOMPOSICIÓN:

dividir un problema o sistema complejo en partes más pequeñas y manejables.



PENSAMIENTO ALGORÍTMICO:

desarrollar una solución paso a paso para resolver un problema.



PATRONES:

identificar objetos o comportamientos que son recurrentes y clasificarlos.

Los desafíos Bebras presentan diferentes niveles de complejidad. Los hemos dividido en tres categorías:



Nivel Caral

Este nivel incluye a los estudiantes de 3° y 4° grado de primaria.



Nivel Kuelap

Este nivel incluye a los estudiantes de 5° y 6° grado de primaria.



Nivel Machu Picchu

Este nivel incluye a los estudiantes de 1° y 2° grado de secundaria.



Desafío #7 Pila de ropa



Bruno está ordenando su ropa en una pila sobre la mesa.
Esta ropa la usará al día siguiente en una fiesta.
Las prendas son las siguientes:

Camisa	Polo	Pantalón	Calzoncillo	Tirantes	Medias	Zapatos

Bruno siempre se viste en secuencia comenzando por el artículo más alto de la pila. A Bruno le gusta toda su ropa pero de ninguna manera va a usar los tirantes debajo de una camisa, ni la camisa debajo del polo.

¿Cuáles de las siguientes pilas están en el orden correcto y lista para que Bruno pueda vestirse?

A	B	C	D

Respuesta:

Las respuestas correctas son: B y D

Para encontrar la solución debemos analizar las pilas empezando por el elemento más alto y verificar que el orden es correcto de acuerdo con las restricciones (es decir, las medias deben colocarse antes que los zapatos, la ropa interior debe ir antes que los pantalones, etc.).

La respuesta "A" es incorrecta porque la camisa se pondrá antes que el polo.

La respuesta "C" es incorrecta porque los tirantes están antes que la camisa.

¿Por qué es pensamiento computacional?

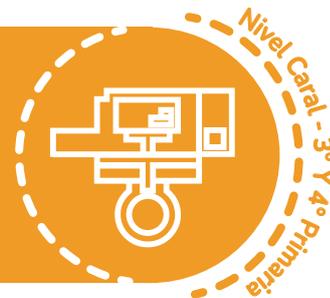
La solución de esta tarea se puede encontrar fácilmente observando la secuencia en la que se encuentran las prendas y verificando si cumple con las limitaciones. Tenemos un conjunto de elementos para vestir y un conjunto de dependencias (es decir, restricciones de "precedencia"), donde "el artículo A debe ejecutarse antes que el artículo B". Una clasificación topológica es un gráfico que permite representar fácilmente las dependencias dadas, especialmente cuando solo se requiere un orden parcial (no todos los elementos están en una relación de dependencia): por ejemplo, puede ponerse las medias antes de ponerse el polo, pero no puede ponerse los zapatos antes que las medias. Gracias a la clasificación topológica, podemos detectar fácilmente las pilas incorrectas.





Desafío #8

La pizza y el tenedor



Lucía está aprendiendo a usar un tenedor.
Su papá le explica que las reglas para comer pizza son:
Los trozos con borde deben tomarse con las manos.
Los trozos sin borde deben tomarse con un tenedor



Marca todos los trozos que Lucía debería comer con su tenedor.



Respuesta:



¿Por qué es pensamiento computacional?

Para cada una de los trozos Lucía debe realizar una prueba simple de si tiene borde o no.

Al diseñar programas para las computadoras, a menudo hay puntos que requieren que el programa elija entre varias opciones.

En programación, una decisión como ésta se conoce como condicional y se implementa mediante instrucciones “Si ... Entonces...”.

Ejemplo: Si llueve, Entonces uso paraguas.



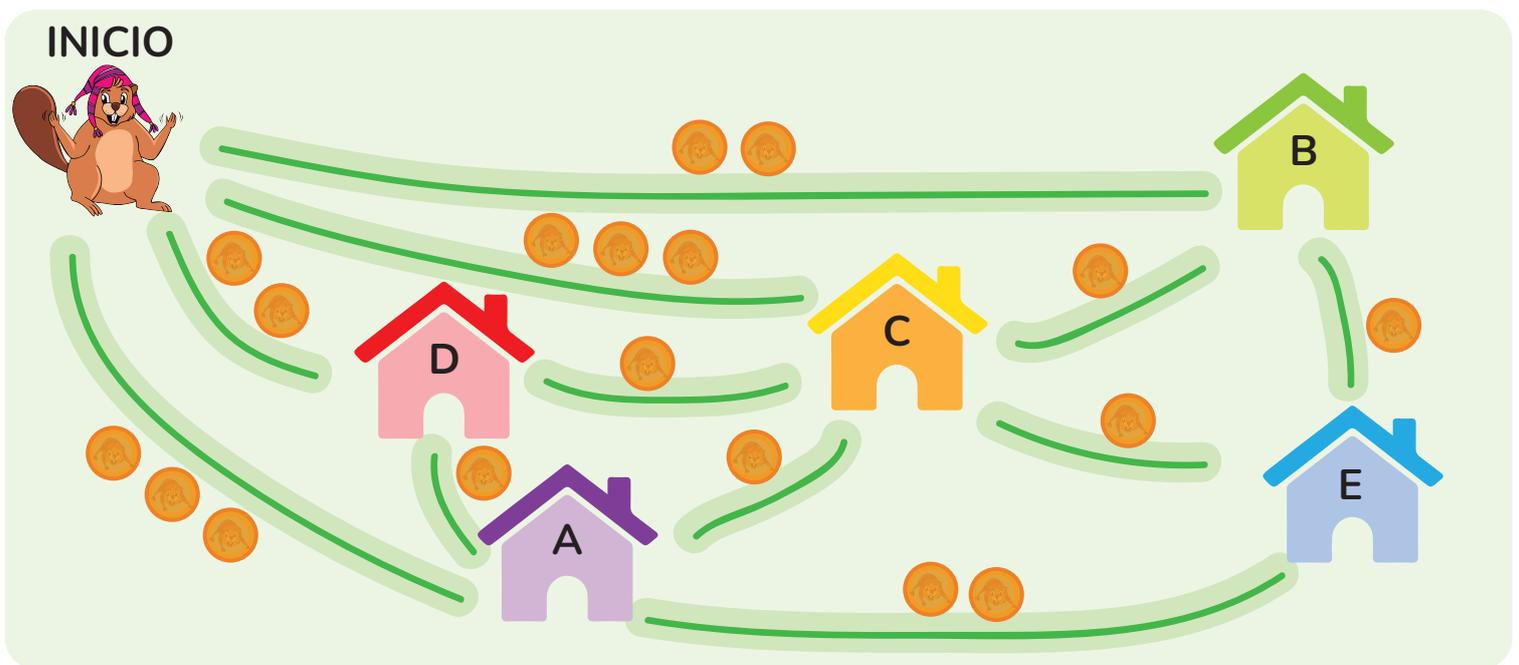


Desafío #7

El camino más barato



Un castor quiere visitar a todos sus amigos en los pueblos A, B, C, D, E en transporte público. Va a visitar a todos sus amigos en un viaje, sin visitar un pueblo más de una vez, y regresando a casa al final del viaje. El precio de cada línea se muestra con monedas en cada trayecto.



Un posible recorrido para visitar a sus amigos que cuesta 11 monedas es:
Inicio -> B -> E -> A -> D -> C -> Inicio.

Encuentra un orden de visitas que le cueste al castor la menor cantidad de monedas posible. Pueden existir múltiples soluciones.

Respuesta:

Hay dos soluciones óptimas:
Inicio -> B -> E -> C -> A -> D -> Inicio
Inicio -> D -> A -> C -> E -> B -> Inicio

Las dos soluciones son inversas y ambas cuestan 9 monedas. No hay una mejor solución que otra, ya que por ejemplo si comienzas desde la casa del castor por el camino de las dos monedas (Inicio -> B), debes ir por el camino de las tres monedas al final, lo que suma la misma cantidad de monedas si lo haces a la inversa. Visitar los otros nodos son 4 formas que cuestan una moneda por cada tramo, lo que da un total de 9 monedas. El resto de las soluciones cuestan más de 9 monedas.

¿Por qué es pensamiento computacional?

Buscar soluciones buenas o incluso óptimas es una de las tareas fundamentales de la computación. Podemos visualizar la descripción de esta tarea de optimización en un gráfico donde los pueblos de los amigos son nodos y las rutas son bordes. La tarea es visitar todos los nodos exactamente una vez de tal manera que la suma de los el peso de los bordes (el costo en monedas de castor) es mínimo.

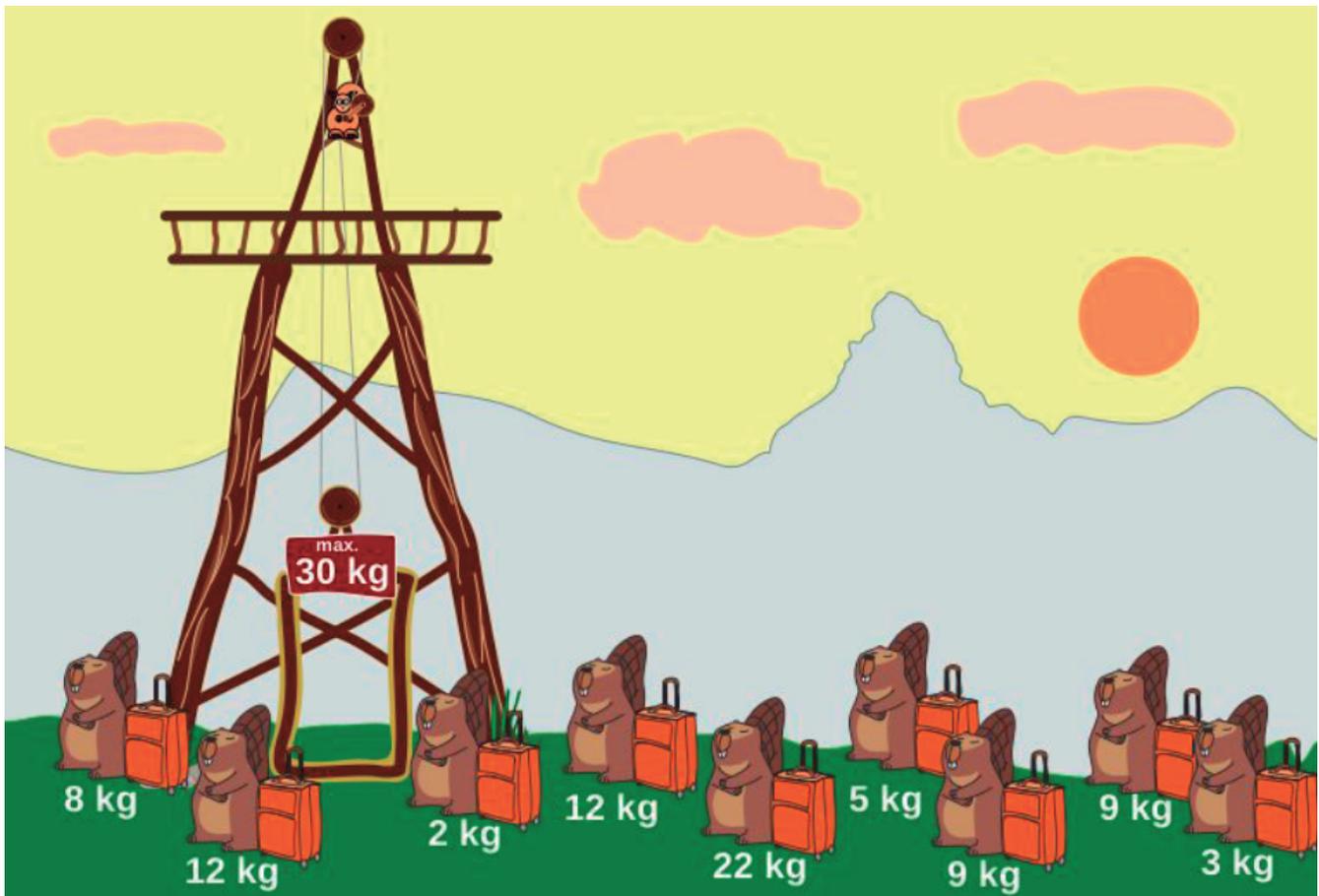




Desafío #8 El ascensor



Un grupo de castores está visitando un parque y quiere tomar el ascensor hasta el mirador, pero es tarde y el ascensor solo sube dos veces. El ascensor tiene una capacidad de carga de 30 kg.



¿Cómo podemos distribuir a los castores con su equipaje entre los dos viajes del ascensor para que suban tantos castores como sea posible?

Respuesta:

La primera idea podría ser que ingresen en el primera viaje la mayor cantidad posible de castores con el menor Peso: $2 + 3 + 5 + 8 + 9 = 27$ kg y en el segundo viaje $9 + 12 = 21$ kg (7 castores en total). Pero es posible que ingresen 8 castores en los ascensores:



La estrategia: intercambiar el castor de 9 kg por el castor de 12 kg en el primer ascensor (peso total del primer ascensor 30 kg), así puede ingresar el castor de 9 kg en el segundo ascensor (peso total del segundo ascensor 30 kg).

¿Por qué es pensamiento computacional?

En este problema nos enfrentamos a demasiadas posibilidades, y es imposible verificarlas todas de manera razonable. Tenemos que encontrar la 'mejor solución posible' al problema, sin embargo, esta no siempre puede ser la solución óptima. En computación se habla entonces de un problema prácticamente irresoluble. Sin embargo seguimos una plan inteligente, comenzando por tratar de colocar tantos castores como sea posible en el primer ascensor. Ese plan que conduce a una solución buena, pero posiblemente no la mejor, se denomina heurística.

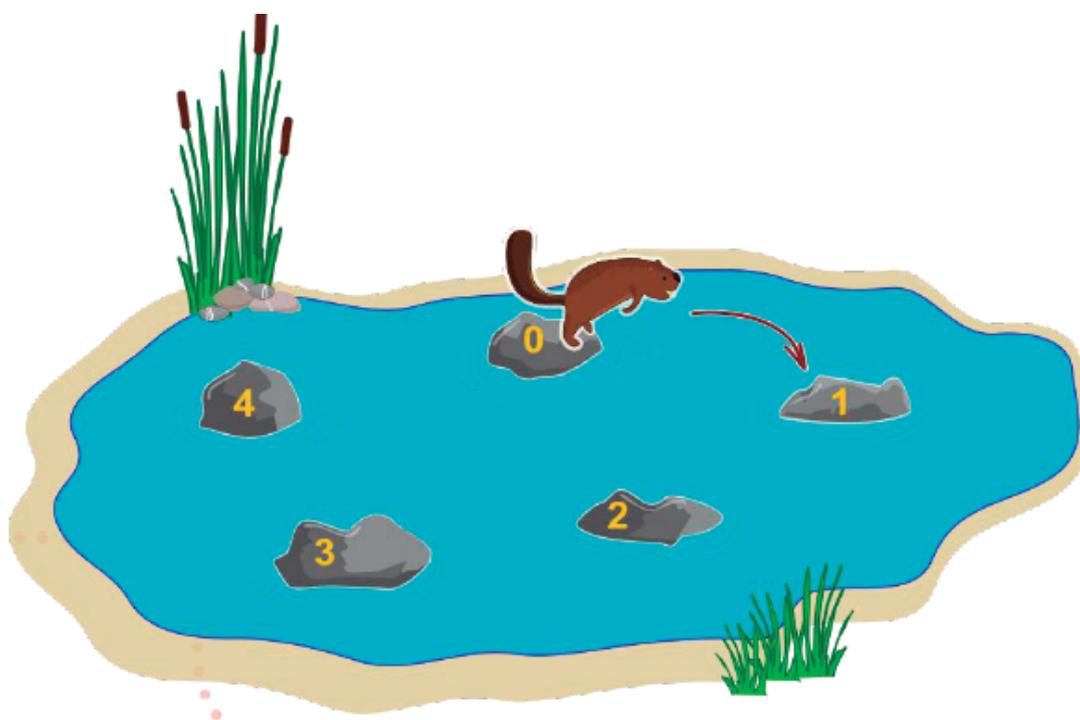




Desafío #6 Módulo de castor



En un campeonato anual de castores, los participantes se encuentran con el primer desafío. Este consta de saltar de roca en roca en el sentido de las agujas del reloj como lo muestra la figura comenzando desde la roca número 0.



Entonces, si un castor salta 8 veces, terminará en la roca número 3:

0 -> 1 -> 2 -> 3 -> 4-> 0 -> 1 -> 2 -> 3

**Uno de los castores saltó 129 veces.
¿En qué roca terminó?**

Respuesta:

Si un castor salta 5 veces termina donde está. Llamémoslo una "vuelta". Para saber dónde termina después de 129 saltos tenemos que averiguar cuántas "vueltas" salta y cuántos saltos le quedan por hacer después de eso.

En este caso es $129 = 25 \times 5 + 4$. Entonces, al saltar 129 veces terminará en el mismo lugar que si hubiera saltado 4 veces. Termina en la piedra número 4.

¿Por qué es pensamiento computacional?



En este caso, se le pide que calcule el resto de $129 \div 5$. Dado que esta operación se usa muy comúnmente en las computadoras, tiene un nombre: operación de módulo. Por lo general se utiliza "%" o "mod" como operador. Así que para nuestra ecuación podríamos escribir: $129 \% 5 = 4$.

Las aplicaciones típicas de este operador son en bucles de programas (al igual que nuestro castor saltando en bucles) cuando las variables se desbordan o incluso para el criptosistema RSA generalizado.



Desafío #7 El código secreto



José tiene una cueva donde ha escondido todos sus tesoros. Desafortunadamente, ha olvidado el código de 3 dígitos de la alarma. Previendo que esto podría pasar, se escribió las siguientes pistas:

- Pista 1: 1 7 2 Solo uno de estos dígitos es correcto, pero está en la posición incorrecta.
- Pista 2: 8 5 4 Dos de estos dígitos son correctos, pero están en la posición incorrecta.
- Pista 3: 9 8 6 Solo uno de estos dígitos es correcto, pero está en la posición incorrecta.
- Pista 4: 7 5 1 Solo uno de estos dígitos es correcto y está en la posición correcta.

¿Cuál es el código correcto?



Respuesta:

De acuerdo con la Pista 1 y la Pista 2, el código puede contener dígitos de la siguiente lista: 1 7 2 8 5 4

Según la Pista 3, 8 es el único dígito que es común entre ese y la lista original (1 7 2 8 5 4). La posición del 8 debe ser la tercera posición porque la Pista 2 y la Pista 3 nos dicen que el 8 está en la posición incorrecta.

Sabemos que el dígito 8 es correcto y está en la posición correcta. De acuerdo con la Pista 1 y la Pista 4 hay un dígito correcto, en la Pista 1 está en la posición incorrecta y en la Pista 4 está en la posición correcta. El único dígito común es 7 y la Pista 4 nos dice que está en la posición correcta. La Pista 2 nos dice que hay un dígito correcto y el único dígito restante es 4.

Por lo tanto el código es 7 4 8

¿Por qué es pensamiento computacional?

Esta tarea se relaciona con la lógica: las sugerencias son restricciones lógicas que toda solución correcta debe satisfacer.

Vale la pena señalar que en este caso la solución es única, por lo que no es una buena manera de "ocultar" un código de contraseña.

Conocer las pistas es lo mismo que conocer el código en sí.

