

TAREA #8

En esta tarea tendrás el reto de resolver uno de los problemas clásicos que pueden ser resueltos con la técnica de la **Programación dinámica**. El objetivo es que practiques la estrategia que se sigue para resolver problemas con esta técnica. Dado que es un problema clásico, puedes encontrar muchas soluciones en Internet, pero evita caer en la tentación de copiarlas. Haz el esfuerzo por enfrentar el caso con tus habilidades, y apóyate con información de Internet que realmente te haga razonar y apropiarte de la solución. En esta ocasión, podrás trabajar en equipo con un compañero con el que estés de acuerdo en colaborar. Entre ambos, deberán resolver el problema con las variantes que se indican. No es la idea dividirse el trabajo y después integrar soluciones, sino que ambos, aportando ideas y experiencia, resuelvan todo y aprendan de igual manera. Esta es una oportunidad de practicar el trabajo colaborativo. Si no tienes la disposición para esto, es mejor que permanezcas sólo en la solución de la tarea, y no afectes a algún compañero. La evidencia final del aprendizaje es individual y será a través de un video en el que expliques la solución y el aprendizaje logrado.

CASO A RESOLVER

Las máquinas automáticas expendedoras de productos son muy comunes en múltiples lugares, al igual que los cajeros automáticos para el pago de estacionamientos. Estas máquinas en la mayoría de los casos tienen en común la función de regresar el cambio de un pago con monedas. Para hacerlo, consideran regresar el mínimo de monedas que conforman la cantidad que se tiene como cambio. Detrás de esto, hay un algoritmo que permite que la máquina regrese las monedas adecuadas. Este algoritmo corresponde a lo que se ha identificado como el **problema del cambio de monedas** (coin change).

En forma concreta, el problema se describe de la siguiente manera: *Dado un conjunto de n tipos de monedas, cada una con valor v_i , y dada una cantidad P , encontrar el número mínimo de monedas que tenemos que usar para obtener la cantidad P .*

Una suposición para resolver inicialmente el problema es que se cuenta con un número ilimitado de monedas de cada tipo, pero en la situación real de una máquina, se tendría físicamente un número concreto de monedas de cada tipo, y esto sería un dato de entrada también para el algoritmo.

Realiza la implementación de un programa en C++ que dada una cantidad a dar como cambio, indique cuál es el mínimo de monedas que conformarían el cambio, utilizando las monedas mexicanas que actualmente circulan (sin considerar las de centavos) y que son las siguientes:



La primera versión del programa deberá funcionar considerando que se tiene un número ilimitado de monedas de cada tipo para dar como cambio.

Una vez que tengas funcionando la versión anterior, deberás adaptarla para generar la segunda versión en la que el usuario dará también como entrada cuántas monedas de cada tipo están disponibles para dar el cambio, y en base a eso dará el resultado.

Una última versión del programa considerará la adaptación sobre la versión anterior para que el programa indique cuántas monedas de cada tipo conforman el cambio.

Estrategia de solución

Este problema puede ser resuelto con diversas técnicas de diseño de algoritmos, pero deberás utilizar la **Programación dinámica** para solucionarlo en esta tarea.

El primer paso será diseñar la definición de la fórmula recursiva que permite encontrar la solución al problema. Para hacer esto, considera las condiciones del programa en su versión #1. Aunque hay muchas explicaciones en Internet en las que te puedes apoyar, intenta hacer el diseño sin ver soluciones para que ejercites esta capacidad de diseño, pero en caso de que lo necesites, apóyate con las explicaciones que consideres valiosas. Lo importante es que te asegures que comprendes muy bien la fórmula correspondiente.

Una vez que se tenga la fórmula, se deberá decidir cuál es el espacio de memoria que se requiere para almacenar resultados en el proceso *bottom-up* del algoritmo. Al igual que con la fórmula, trata de hacer tú el diseño, pero apóyate con las explicaciones de Internet en caso necesario.

Finalmente, con los diseños anteriores, puedes pasar a diseñar el algoritmo y programarlo para la versión #1.

La versión #2 implicará considerar la restricción de la cantidad de las monedas en la fórmula recursiva y en el diseño del algoritmo. Realiza los cambios necesarios y documéntalos.

Para la versión #3 tendrás que idear una manera de identificar las monedas que conforman el cambio óptimo de la versión #2. Apóyate en referencias de Internet para diseñar esta parte y construir el algoritmo a implementar.

Condiciones para la entrega de resultados

La entrega de esta tarea deberá ser INDIVIDUAL considerando lo siguiente:

1. Entregar en forma impresa el código de las 3 versiones de programas realizados. Los códigos deberán estar documentados con comentarios que indiquen los cambios en cada versión, las fórmulas recursivas utilizadas, el diseño de los espacios de memoria para resultados parciales.
2. Grabar un video en el que se narre la historia del proceso de solución al caso, explicando los diseños realizados y mostrando la ejecución de los programas con sus pruebas. Se deberá mencionar cómo fue el trabajo en equipo, indicando cuál fue la principal aportación en la solución de la tarea de cada integrante. Se deberá comentar acerca del aprendizaje logrado y dar una opinión personal del uso de la técnica de la programación dinámica. Este video se subirá como siempre en la página de Fb del curso y no deberá excederse a 5 minutos.
3. Este caso es equivalente al problema 322 (*Coin Change*) en la plataforma *leetcode*, por lo que deberán trabajarlo también en esta plataforma y entregar evidencia de la aceptación del mismo a través de una impresión de la pantalla correspondiente.

Se espera que esta tarea se entregue el **martes 26 de marzo** para ser calificada sobre 100. Será calificada sobre 80 puntos en caso de que se entregue entre el miércoles 27 y el viernes 29 de marzo. A partir de esta última fecha, cada día de retraso penalizará con 10 puntos la calificación. La calificación podrá ser diferente para cada integrante del equipo, dependiendo de las evidencias que muestren en la entrega.

La entrega completa y correcta de esta tarea será considerada como un problema resuelto (medium) del momento de práctica en la ronda #3 del reto semestral.

Para cualquier duda o asesoría, consulta al profesor.