

Análisis y diseño de algoritmos

Cambio exacto con monedas limitadas

July 17, 2025

Descripción del proyecto

Dado un conjunto de denominaciones de monedas y un número máximo de monedas disponibles por cada tipo, determinar todas las formas posibles de alcanzar una cantidad objetivo exacta utilizando una o más monedas, sin exceder los límites establecidos.

La solución debe encontrarse usando backtracking y cada combinación válida debe mostrarse como un vector indicando cuántas monedas de cada denominación se utilizaron.

Ejemplo:

- Denominaciones: $\{1, 2, 5\}$
- Cantidad objetivo: 5
- Límite por moneda: $\{3, 2, 1\}$

Posibles soluciones:

- $\{0, 0, 1\} \rightarrow 5$
- $\{1, 2, 0\} \rightarrow 1 + 2 + 2 = 5$
- $\{3, 1, 0\} \rightarrow 1 + 1 + 1 + 2 = 5$

Formato de entrada

El archivo de entrada contiene múltiples casos de prueba. Cada caso incluye:

- Una línea con un entero n ($1 \leq n \leq 20$) indicando el número de denominaciones.
- Una línea con n enteros representando las denominaciones (positivos y en orden creciente o arbitrario).
- Una línea con n enteros indicando el número máximo disponible de monedas por tipo.
- Una línea con un entero T indicando la cantidad objetivo a alcanzar.

Formato de salida

Para cada caso, mostrar todas las combinaciones posibles de monedas que suman exactamente T sin exceder los límites.

También puede implementarse una extensión opcional que muestre únicamente la combinación con el menor número total de monedas utilizadas.

Pseudocódigo

Algoritmo CAMBIOEXACTO(pos , $acumulado$)

```
1 si  $acumulado > T$  entonces
2   |   regresar
3 fin
4 si  $pos = n$  entonces
5   |   si  $acumulado = T$  entonces
6   |   |   escribir combinación actual
7   |   fin
8   |   regresar
9 fin
10 para  $i \leftarrow 0$  hasta  $limite[pos]$  hacer
11   |   Asignar  $i$  monedas de  $denominacion[pos]$ 
12   |    $combinacion[pos] \leftarrow i$ 
13   |   CAMBIOEXACTO( $pos + 1$ ,  $acumulado + i \cdot denominacion[pos]$ )
14 fin
```

El algoritmo emplea la técnica de backtracking para explorar todas las combinaciones posibles de monedas, respetando los límites disponibles por denominación y evitando soluciones parciales que superen el total objetivo.

En cada paso, se elige una cantidad válida de monedas para la denominación actual (desde 0 hasta el máximo permitido) y se avanza recursivamente a la siguiente. Si en algún punto la suma acumulada excede el objetivo, la rama se descarta. Si se alcanza el final y la suma es exactamente igual al objetivo, la combinación se reporta como válida.

El arreglo **combinacion**[] se utiliza para almacenar temporalmente las cantidades elegidas de cada tipo y es sobrescrito en cada nivel de la recursividad (por lo tanto, no se requiere deshacer los cambios manualmente).

Entrega

- Código fuente implementando la solución.
- Informe breve explicando la solución y los resultados.
- Presentación a utilizar en la exposición del proyecto.

- Exposición del proyecto en la semana 11 del trimestre.
- Entrega a más tardar el 31 de julio a las 12:00 del mediodía.

Evaluación

Este proyecto forma parte de un **torneo académico**, en el que todos los equipos compiten por obtener la mejor calificación posible. Las soluciones serán evaluadas bajo un esquema **relativo y comparativo**, tomando como referencia las mejores implementaciones, explicaciones y presentaciones.

Componentes de la evaluación

- **Informe técnico (60%)**: Se evaluará la corrección del algoritmo, el uso apropiado de backtracking, la eficiencia, el manejo de archivos, los casos de prueba y la claridad y documentación del código, conforme a la rúbrica general.
- **Presentación oral (40%)**: Se evaluará la claridad en la exposición del problema, el dominio del algoritmo, la justificación de las decisiones, el uso de recursos visuales y la participación del equipo.

Participación del público

Durante cada presentación, los demás equipos deberán asumir el rol de **jurado y oponentes técnicos**. Esto significa que:

- Pueden realizar preguntas desafiantes sobre la solución, decisiones de diseño, manejo de casos extremos, eficiencia, etc.
- Si un equipo no logra responder adecuadamente una pregunta importante del público, su calificación podrá verse afectada dentro del criterio de dominio del algoritmo.
- Por el contrario, si un equipo responde con solidez y claridad, podrá reforzar su calificación.
- Se premiará simbólicamente la **mejor pregunta del público**, especialmente si logra evidenciar una debilidad relevante en la solución presentada.

Se fomentará un ambiente competitivo pero respetuoso, donde el objetivo es aprender más a través del reto mutuo y el análisis crítico entre pares.

Notas

- Se recomienda probar el programa con los conjuntos proporcionados.
- El algoritmo debe ser eficiente y evitar soluciones redundantes.

- El uso de funciones auxiliares para validar la selección de denominaciones es obligatorio.
- El alumnado está sujeto a una entrevista para comprobar la autoría de sus trabajos.