

Análisis de problema 1.

El problema consiste en encontrar la cantidad mínima de fresas en una fila continua de tarros de mermelada. Se puede resolver utilizando el algoritmo de programación dinámica "Max 1D Range Sum".

Este algoritmo funciona dividiendo el problema en subproblemas más pequeños. Cada subproblema consiste en encontrar la cantidad mínima de fresas en una fila continua de tarros de mermelada, comenzando en un tarro dado.

La solución a un subproblema depende solo de las soluciones a los subproblemas anteriores. En este caso, la solución a un subproblema depende solo de las diferencias de fresa en los tarros anteriores al tarro en el que comienza el subproblema.

El algoritmo funciona recorriendo todos los subproblemas y manteniendo dos variables: *minSum* y *currentSum*. La variable *minSum* representa la suma mínima de fresas en una fila continua hasta el momento, y la variable *currentSum* representa la suma actual de fresas en una fila continua.

La función actualiza *currentSum* agregando el valor del tarro actual. Si la suma actual es menor que la suma mínima, actualiza la suma mínima. Si la suma actual es positiva, la reinicia.

La función devuelve la suma mínima, que es la solución al problema.

En este caso, la función recorre todos los tarros de mermelada, uno por uno. Para cada tarro, la función calcula la cantidad mínima de fresas en una fila continua que comienza en ese tarro. La función utiliza la variable *minSum* para mantener la suma mínima de fresas encontrada hasta el momento.

La solución al problema se encuentra en la variable *minSum*.

Pseudocódigo:

Función `minStrawberries(jars[], N)`:

`minSum = 0`

`currentSum = 0`

Para cada tarro en la fila:

`currentSum += jars[i]`

Si `currentSum < minSum`, `minSum = currentSum`

Si `currentSum > 0`, reiniciar `currentSum` a 0

Devolver `minSum`

Función `main()`:

Leer `T` // Número de casos de prueba

Mientras `T` sea mayor que 0:

Leer `N` // Número de tarros de mermelada

Declarar arreglo `jars` de tamaño `N`

```
Para cada i desde 0 hasta N-1:  
    Leer jars[i] // Diferencia de fresa en el tarro i  
  
result = minStrawberries(jars, N)  
Mostrar "Cantidad mínima de fresas en una fila continua:", result  
  
Decrementar T en 1
```

Ejecución manual

Supongamos que tenemos los siguientes tarros de mermelada:

```
jars = [-2, 1, 3, -4, 5]
```

Ejecución manual:

Paso 1: Inicializamos las variables minSum y currentSum.

```
minSum = 0  
currentSum = 0
```

Paso 2: Recorremos los tarros de mermelada, uno por uno.

```
for i in range(len(jars)):
```

Paso 3: Actualizamos la variable currentSum agregando el valor del tarro actual.

```
currentSum += jars[i]
```

Paso 4: Si la suma actual es menor que la suma mínima, actualizamos la suma mínima.

```
if currentSum < minSum:
```

```
    minSum = currentSum
```

Paso 5: Si la suma actual es positiva, reiniciamos la suma actual.

```
elif currentSum > 0:
```

```
    currentSum = 0
```

Paso 6: Devolvemos la suma mínima.

```
return minSum
```

Ejecución del caso:

```
minStrawberries(jars)
```

Salida:

-2

Explicación del código C++ elaborado:

El código C++ implementa el algoritmo de programación dinámica "Max 1D Range Sum" (Máxima Suma en un Rango 1D). La función *minStrawberries* toma como entrada una matriz de diferencias de fresa en cada tarro (*jars*) y el número de tarros (*N*). La función recorre todos los tarros y mantiene dos variables: *minSum* y *currentSum*. La variable *minSum* representa la suma mínima de fresas en una fila continua hasta el momento, y la variable *currentSum* representa la suma actual de fresas en una fila continua. La función actualiza *currentSum* agregando el valor del tarro actual. Si la suma actual es menor que la suma mínima, actualiza la suma mínima. Si la suma actual es positiva, la reinicia. Finalmente, la función devuelve la suma mínima.

La función *main* lee el número de casos de prueba, el número de tarros de mermelada y las diferencias de fresa en cada tarro. Luego, llama a la función *minStrawberries* para calcular la cantidad mínima de fresas en una fila continua y muestra el resultado.