[**L - Smash the Trash**](https://vjudge.net/problem/Gym-103430L)

**Reducción del Problema**

**Descripción en palabras:**

Queremos calcular el número mínimo de personas necesarias para limpiar una calle con basura en diferentes posiciones. Cada persona puede hacer una de tres cosas por día: eliminar 1 kilogramo de basura en una posición, mover 1 kilogramo de basura de una posición a la siguiente, o no hacer nada. La calle se considera limpia cuando ya no queda basura al final del proceso.

**Descripción en lenguaje matemático:**

Dado un arreglo T = [t1, t2, ..., tn] que representa la cantidad de basura en cada una de las n posiciones, se busca el valor mínimo de m tal que sea posible, usando m personas por día, eliminar o mover toda la basura en un número finito de días, hasta que en todas las posiciones haya cero basura.

**Reducción de la Solución**

**Descripción en palabras:**

La cantidad mínima de personas necesarias corresponde al máximo valor absoluto que se alcanza en la suma acumulada de los cambios de basura entre posiciones consecutivas. Esto se debe a que al mover o eliminar la basura, se generan acumulaciones temporales que definen el punto más crítico de trabajo.

**Descripción en lenguaje matemático:**

Sea acumulador = 0 y respuesta = 0.

Para i desde 1 hasta n:

acumulador += (ti - ti-1)

respuesta = max(respuesta, abs(acumulador))

donde t0 = 0.

Al final, la respuesta es el número mínimo de personas necesarias.

**3. Código Realizado y Análisis**

* Enlace(s) Código: [**https://vjudge.net/solution/61083819/dLjIMVb1BTNi1TYvUICM**](https://vjudge.net/solution/61083819/dLjIMVb1BTNi1TYvUICM)

**4. Casos de Prueba**

**Entrada:**

3

5 1 2

Salida :

4

**Entrada**:

4

8 1 2 7

Salida:

7

**Justificación**:

En el primer caso, se pueden contratar 4 personas para limpiar la calle. En el día 1, tres personas eliminan 1 kg cada una en la posición 1 y una persona mueve 2 kg a la posición 2. En el día 2, se limpian los 3 kg acumulados en la posición 2. En el día 3, se eliminan los 2 kg finales en la posición 3. No se puede hacer con menos de 4 personas porque en algún día se requiere disponer de al menos 4 acciones simultáneas.

En el segundo caso, se requieren al menos 7 personas porque en el último día hay 7 kg de basura en la última posición, y cada persona solo puede eliminar 1 kg por día. Por lo tanto, 7 personas son necesarias como mínimo para limpiar esa posición, sin importar la estrategia previa.

**5. Iteración en Caso de Solución Incorrecta (o explicación Solución Correcta)**

**Código incorrecto:** [**https://vjudge.net/solution/61031918/OIZA7p13fCw5MK7kWXZ8**](https://vjudge.net/solution/61031918/OIZA7p13fCw5MK7kWXZ8)

* **Descripción del error**:

El código falla porque acumula simplemente la suma prefija de todos los kilos de basura:

basura\_acumulada += basura[i]

Pero la solución correcta no es “cuánta basura hay hasta ahora”, sino “cuál es el pico de cargas simultáneas que se forman al mover o eliminar basura”.

En el enunciado, cada día se puede

* eliminar 1 kg en la posición i, o
* mover 1 kg de i a i+1

y ese movimiento hace que la “carga de trabajo” siga un perfil de diferencias, no de sumas directas.

el acumulador tal como está, para el caso  
 5 1 2  
 tu basura\_acumulada sería  
 día 1 → 5  
 día 2 → 5+1=6  
 día 3 → 6+2=8  
 y por eso imprimes 8, cuando la respuesta real es 4.

* **Proceso de depuración:**

Primero repasé el enunciado y confirmé que necesitamos decidir si, dado un número fijo de trabajadores m, éstos pueden limpiar la calle respetando las reglas de eliminar o mover 1 kg por día cada uno. A partir de ahí decidí usar búsqueda binaria sobre m, con límites iniciales entre 0 y la suma total de basura S.

Luego escribí la función es\_factible(t, m): recorro cada posición i llevando un “carry” que representa la basura arrastrada desde la posición anterior. Calculo total = t[i] + carry; si total ≤ m, eso significa que con m trabajadores puedo eliminar todo y no queda basura para mover; pongo carry = 0. Si total > m, compruebo que total ≤ 2·m (porque para mover kilos al siguiente sitio cada kilo movido requiere dos acciones: extraerlo y depositarlo en i+1) y que no sea la última posición; de lo contrario, m no sirve y devuelvo False. Si es válido, determino moves = total – m y actualizo carry = 2·moves para pasarlo al siguiente i. Si completo el bucle, devuelvo True.

Con esa función, en main inicializo lo = 0, hi = S y mientras lo ≤ hi calculo mid = (lo+hi)//2. Si mid = 0 trato el caso especial: si todo t[i] = 0 imprimo 0 y salgo, si no ajusto lo = 1. Para mid > 0 llamo a es\_factible: si es True guardo respuesta = mid y acorto hi = mid–1; si es False, aumento lo = mid+1. Al terminar imprimo la respuesta mínima encontrada.

Durante la depuración probé ejemplos sencillos como [5,1,2] y [8,1,2,7], comprobando en cada iteración de búsqueda binaria que los valores de carry y los retornos de es\_factible coincidían con las simulaciones manuales de “¿puedo mover y eliminar con m trabajadores?”. Ajusté cuidadosamente la condición total > 2\*m para detectar casos imposibles, y verifiqué que en la última posición no permitiera movimientos extra. Con eso confirmé que la búsqueda binaria converge al mínimo m correcto en O(n log S).

* **Solución corregida:**

<https://vjudge.net/solution/61045391/1zakO5XuoQYKkVovzDMZ>

**6. Preguntas de Aprendizaje**

* **Temas aplicados:**
* Búsqueda binaria
* Función de factibilidad
* Técnica del acarreo
* Uso de diferencias acumuladas implícitas
* **Dificultad de la implementación**: Media, bueno fue un poco difícil, porque no sabia eso de la técnica de acarreo(osea si sabía que se podía usar pero no recordaba cómo implementarla)así tuve que buscar como se usa.

* **Recursos utilizados:**

<https://medium.com/%40alok.g.v/data-structures-and-algorithms-dsa-concept-carry-forward-5e753b964207>

**7. Feedback LLM**

* **Envío código LLM**: <https://vjudge.net/solution/61083873/0dfn5WD5hIlzE3UNPZ2j>
* **Comparación de su códigos**:

El código LLM se parece a la primera versión mía, solo le dije el problema y que lo resolviera no le dije como hacerlo ni nada asi que penso lo mismo que yo pensé al principio osea la suma prefija de todos los kilos de basura, ya la versión final mia usar la técnica Carry forward, búsqueda binaria etc, eso me hace pensar que para problemas específicos si uno no usa bien la IA no los podía resolver porque me paso en varios ejercicios en este taller.

* **Feedback de LLM**: Pida al LLM que evalúe y corrija su código:







