Cultivando plantas (cultivando)

Enunciado

Un agricultor tiene M tipos de plantas que quiere cultivar en su invernadero durante N días.

- Cada tipo de planta i necesita exactamente a_i días de preparación (regar, fertilizar, limpiar la tierra) antes de poder plantarla.
- El calendario de siembra es especial: cada día puede estar disponible la oportunidad de plantar cierto tipo de planta. En otros días, no hay oportunidad de plantar nada.
- En un día cualquiera, el agricultor solo puede realizar **una acción**:
  - preparar la tierra para alguna planta (esto cuenta como un día de preparación para esa planta),
  - plantar la planta correspondiente (si ese día está disponible y la preparación ya se cumplió),
  - o descansar (aunque no es recomendable).

El objetivo es plantar todas las plantas (cada tipo exactamente una vez).

Debes determinar el mínimo número de días (longitud de un prefijo del calendario) necesarios para lograrlo, o indicar que es imposible.

Implementación

Debes implementar la siguiente función en C++:

```
int minimo_dias_para_plantar(
  int N, std::vector<int> d, int M, std::vector<int> a
);
```

O la siguiente función en Python:

```
def minimo_dias_para_plantar(
  N: int, d: list[int], M: int, a: list[int]
) -> int:
```

Parámetros

- `N`: número total de días en el calendario.

- `d`: arreglo de longitud N. $d[t] = 0$ indica que el día $t + 1$ no se puede plantar nada; si $1 \leq d[t] \leq M$, indica que el día $t + 1$ se puede plantar la planta de tipo $d[t]$.
- `M`: número total de tipos de plantas.
- `a`: arreglo de longitud M. $a[i - 1]$ es la cantidad de días de preparación que requiere la planta de tipo i.

Retorna

- El menor X tal que usando únicamente los días del 1 al X es posible plantar todos los tipos de plantas cumpliendo las reglas.
- Si no es posible plantar todas las plantas en ningún prefijo, retorna `-1`.

Ejemplos

Ejemplo 1

En la siguiente llamada:

```
minimo_dias_para_plantar(7, [0, 1, 0, 2, 1, 0, 2], 2, [2, 1])
```

Debe retornar `5`

Ejemplo 2

En la siguiente llamada:

```
minimo_dias_para_plantar(
  10, [0, 0, 1, 2, 3, 0, 2, 0, 1, 2], 3, [1, 1, 4]
)
```

Debe retornar `9`

Ejemplo 3

En la siguiente llamada:

```
minimo_dias_para_plantar(5, [1, 1, 1, 1, 1], 1, [6])
```

Debe retornar `-1`

Límites

- $1 \leq N, M \leq 10^5$
- $1 \leq a_i \leq 10^5$

Subtareas

Subtarea 1 (4 puntos)

- $M = 1$

Subtarea 2 (6 puntos)

- $a_i = 1$, es decir cada planta requiere un solo día de preparación.

Subtarea 3 (13 puntos)

- En el arreglo d, cada planta aparece **exactamente una vez**.

Subtarea 4 (15 puntos)

- En el arreglo d, Todos los `0` (días libres) aparecen al inicio, seguidos únicamente de días de siembra (posiblemente repetidas).

Subtarea 5 (25 puntos)

- $1 \leq N, M \leq 20$
- $1 \leq a_i \leq 20$

Subtarea 6 (37 puntos)

- Sin restricciones adicionales al problema original.

Entrada

- Solo debes subir el archivo `cultivando.cpp` con tu solución.
- La entrada y salida serán manejadas por el archivo `grader.cpp` o `grader.py`.
- **No debes imprimir nada por la salida estándar** ni leer de la entrada estándar en tu archivo `cultivando.cpp`.
- No modifiques los archivos `grader.cpp` o `grader.py`, ya que el juez solo tomará en cuenta tu solución en `cultivando.cpp`.