

UNIVERSIDAD DE GUANAJUATO
DIVISIÓN DE CIENCIAS NATURALES Y EXACTAS
CAMPUS GUANAJUATO

Tarea 3 (Estructura de datos y algoritmos)

Nombre: Ricardo León Martínez

Fecha: 13/2/2026

Calificación: _____

1. Problema A

la solucion no intenta simular segundo por segundo porque t puede ser muy grande. En lugar de eso, aplica busqueda bianria sobre el tiempo para encontrar el mínimo número de segundos necesarios para producir t productos. Si en cierto tiempo T se pueden fabricar t productos, entonces en cualquier tiempo mayor se podra. Esto hace un patron donde primer hay tiempos que no alcanzan la meta y luego, a partir de cierto punto, todos los tiempos sí la alcanzan. Ese cambio permite usar busqueda binaria. El programa lee n, t y los tiempos de cada maquina. Luego define un rango grande de tiempo posible y en cada iteración prueba el punto medio. Calcula cuántos productos se pueden fabricar en ese tiempo sumando $mitad/k[i]$ para cada maquina. Si alcanza al menos t productos, guarda ese tiempo como posible respuesta y busca uno menos. Si no, busca uno mayor. Así, mediante busqueda binaria, encuentra el menor tiempo que permite producir los t productos.

2. Problema B

La idea es muy similar al problema anterior, en lugar de probar todas las alturas posibles una por una, utilizas busqueda binaria sobre la altura del corte para encontrar la maxima altura que todavía permite obtener al menos M metros de madera. Aquí no buscamos el mínimo valor que funciona, sino el máximo valor que aún cumple la condición. Si fijamos H muy baja, se corta mucha madera. Si vamos aumentando H , la cantidad de madera obtenida disminuye. Es decir, a mayor altura de corte, menor madera obtenida. Es decir, primero alturas que si permiten obtener al menos M metros, y despues alturas que ya no alcanzan. Eso es lo que nos permite usar la busqueda binaria. El programa lee N, M y las alturas de los arboles. Luego define un rango de busqueda entre 0 y 10^9 . En cada iteracion calcula la altura media del intervalo determina cuanta madera se obtiene si se corta a esa altura. Para cada arbol, si su altura es mayor que la altura de corte, se suma la diferencia $h[i] - mitad$. Si en algún momento la madera acumulada alcanza M , se detiene el cálculo. Si con esa altura al menos M metros, entonces esa altura es válida y se intenta subir más la sierra para maximizar la altura. Si no se alcanza M , significa que la sierra está demasiado alta y se debe bajar.

3. Problema C

Aquí la idea es búsqueda binaria sobre la respuesta, pero ahora la respuesta es la penalización mínima posible. Suponemos que la penalización máxima permitida es X y verificamos si es posible cumplir con esa cota usando a lo más k operaciones. Si una celda tiene penalización mayor que X , entonces no podemos permitir que quede mal pintada. En cambio, si su penalización es menor o igual a X , podemos ignorarlo si queda incorrecta, porque el máximo seguiría siendo $\leq X$. Entonces para un valor fijo X , el problema se reduce a asegurar que todas las celdas con $a[i] > X$ queden correctamente pintadas. En la función `possible(X)` se recorre la tira de izquierda a derecha. Si encontramos una celda que debe quedar roja y su penalización es mayor que X , entonces no podemos estar en medio de un segmento azul, así que se corta cualquier segmento activo. Si encontramos una celda que debe quedar azul y su penalización es mayor que X , entonces estamos obligados a pintarla de azul. Si no estamos actualmente pintando un segmento, iniciamos uno nuevo y aumentamos el contador de operaciones. De esta manera, el número de operaciones que realmente necesitamos es el número de bloques obligatorios de celdas azules que tiene penalización mayor que X . Si ese número es menor o igual que k , entonces es posible lograr una penalización máxima $\leq X$. Entonces si cierto X es posible, cualquier valor mayor también lo será, porque estaremos relajando restricciones. Por eso se aplica búsqueda binaria entre 0 y 10^9 . En cada paso se prueba un valor medio. Si es posible, se intenta uno menor para minimizar la penalización, si no, se aprueba uno mayor.