

Técnicas algorítmicas en ingeniería del software

Grado en Ingeniería del Software (UCM)

PRÁCTICA DE PROGRAMACIÓN DINÁMICA

Curso 2014/2015

FECHA LÍMITE DE ENTREGA: 29 DE ENERO

Se valorará tanto la *corrección* como la *eficiencia* y la *simplicidad* de las soluciones propuestas.

El carpintero *Ebanisto* recibe el encargo de cortar un tronco en varios trozos que han sido previamente marcados sobre la madera. El esfuerzo de cortar un trozo de madera en dos es el doble de su longitud. *Ebanisto* se da cuenta de que el orden en el que realice los cortes en el tronco influye en el esfuerzo empleado. Por ejemplo, supongamos que un tronco de longitud 10 metros tiene que cortarse a 2, 4 y 7 metros de uno de los extremos. Una posibilidad sería cortar primero por la marca de los 2 metros, luego por la marca de los 4 metros y finalmente por la de 7 metros, lo que le costaría a *Ebanisto* un esfuerzo de $2 * 10 + 2 * 8 + 2 * 6 = 48$. Sin embargo, si corta primero por la marca del 4, después por la del 2 y finalmente por la del 7, entonces le costaría un esfuerzo de $2 * 10 + 2 * 4 + 2 * 6 = 40$. La práctica consiste en diseñar un algoritmo de programación dinámica que ayude a *Ebanisto* a decidir en qué orden cortar el tronco por las marcas para minimizar el esfuerzo realizado y programarlo.

Nota: Puede interesar también tener “marcas” en los extremos del tronco, aunque ahí nunca se realice un corte.

Entrada

La entrada constará de varios casos de prueba. La primera línea de cada caso contendrá un número positivo l , que representa la longitud del tronco que debemos cortar. La siguiente línea indicará el número de cortes $n > 0$ que se deben realizar. La siguiente línea contendrá n números positivos c_i ($0 < c_i < l$), que determinan los puntos en los que se deben realizar los cortes, dados en orden creciente.

Un caso de prueba en el que $l = 0$ indica el final de la entrada.

Salida

Imprimir el coste de la solución de menor esfuerzo para cortar cada tronco seguido del orden de los cortes, según el formato mostrado en el ejemplo más abajo. Los cortes se muestran en forma de árbol utilizando indentación. Un recorrido en preorden o por niveles de dicho árbol garantiza el coste mínimo.

Ejemplo de entrada

10
3
2 4 7
111
10
10 18 29 31 38 45 48 50 77 94
0

Salida correspondiente

Caso 1: El coste mínimo es 40.
Cortes: 4

2
7

Caso 2: El coste mínimo es 700.
Cortes: 50

29
18
10
38
31
45
48
77
94