Problema B

Ricardo Enrique González

Camilo Zambrano

201425733

201515438

1. *Algoritmo de solución*

Inicialmente se implementó un algoritmo que a diferencia del utilizado, tenía complejidad $T(n^2)$.Se calculaba la credibilidad en cada paso en donde inicialmente se usaba todo el arreglo, luego iba bajando la cota de 1 en 1 reduciendo el tamaño del subarreglo. Cada vez que se conseguía una credibilidad mayor, se almacenaba en un centinela.

En la implementación mejorada, nos basamos en el algoritmo mencionado en la referencia[1] "Kadanes Algorithm". El principio de este problema es conseguir la suma interna de números mayor, en un solo recorrido de un subarreglo. Cada paso se suma el nuevo numero al anterior que se tenia y se compara con un centinela, si se supera, se guarda el nuevo valor mayor y este es el que se imprime.

Se tradujo esta técnica utilizada en el algoritmo, pero se realizaron 2 modificiaciones. El número mayor ahora es la credibilidad y ahora también se almacenan las posiciones en que estaban los índices cuando se calculó la credibilidad mayor.

Básicamente, cada vez que se pasa sobre un número positivo la credibilidad(r) aumenta en 1.
Cada vez que pasa sobre un número negativo la credibilidad disminuye en 1.
En cada paso se pregunta si la nueva credibilidad es mayor a la anterior, si lo es, se guardan los valores actuales r,p,q sobre rcampeon,pcampeon y qcampeon.
Si la credibilidad en algún momento se vuelve negativa, se mueve p que es la variable que representa donde debería comenzar el nuevo posible subarreglo y se resetea el valor de la credibilidad en 0. Esto es, porque la combinación de los elementos anteriores da una credibilidad negativa.
Al final, se imprime los valores que queden en rcampeon,pcampeon y qcampeon.

CTX
$A[0..n-1]$ con $0<n<10^6$
$0 \leq p \leq q \leq n$
$cred(p,q) = (\#i \mid p \leq i < q : A[i]>0) - (\#i \mid p \leq i < q : A[i]<0)$.
Maximizar $cred(p,q)$.

PRE: Se inicializan las 6 variables que van a manejar la respuesta, r,p,q,rcampeon,pcampeon y qcampeon. Todos comienzan en 0 menos q, que comienza en -1. Q se utiliza como contador de posición y los demás se van actualizando de acuerdo a las regla descritas anteriormente. Tambien se inicializa un arreglo de Strings del tamaño de A[0..n-1] que contiene los números a procesar.

POS: En rcampeon queda almacenada la máxima credibilidad obtenida, en pcampeon queda la posición inicial del subconjunto del arreglo y en qcampeon queda la posición final del subconjunto del arreglo. Se imprimen estos 3 valores.

2. *Análisis de complejidades espacial y temporal*

Operaciones realizadas:
Se asume que la lectura y el almacenamiento del string cuesta **O(1)**
Procesar el string obtenido por consola a Vals(arreglo con valores del problema) **O(n)**
Realizar un recorrido de vals para calcular la credibilidad y guardar las posiciones del subarreglo **O(n)**
Imprimir el resultado **O(1)**

$T(n) = O(n+n+1+1) = O(2(n+1)) = O(n)$

Vals: Arreglo de tamaño n que contiene todos los valores procesados **O(n)**
R: variable local que contiene el valor actual de la credibilidad **O(1)**
Rcampeon: centinela de la credibilidad **O(1)**
P: variable local que contiene la posición inicial posible del subarreglo con mayor credibilidad **O(1)**
Pcampeon: centinela que contiene donde comienza el subarreglo con mayor credibilidad **O(1)**
Q: variable local que contiene la posición final posible del subarreglo con mayor credibilidad **O(1)**
Qcampeon: centinela que contiene donde termina el subarreglo con mayor credibilidad **O(1)**

$S(n) = \theta (1+1+1+1+1+1+N) = \theta (6+n) = \theta (n)$

3. *Comentarios finales*

Nuestro algoritmo cumple de manera satisfactoria con todos los casos de que se muestran en el enunciado y otros casos de prueba que consideramos necesario para probar la funcionalidad. Adicionalmente el tiempo de respuesta observado por este parece ser satisfactorio y no se encontraron soluciones más óptimas en complejidad, por tanto consideramos que el algoritmo propuesto es uno de los más adecuados a la solución de este problema.

Nota: Al correr el archivo si no imprime el último valor de inmediato, darle 2 veces a enter.

Referencias:

[1] https://chinmaylokesh.wordpress.com/2011/01/17/for-a-given-array-of-integers-positive-and-negative-find-a-continous-sequence-with-largest-sum/