

Identificación.

Juan Pablo Gonzalez Peña

201424703

Iván Darío Chavarro

201423319

1. Algoritmo de solución.

Suponiendo que se tiene como caso de entrada (2 123542) dicho de otro modo más general la entrada de datos es de la forma k n donde k es el número de iteraciones y n el número que se debe procesar. A continuación podrán ver un ejemplo concreto donde se muestra como funciona el algoritmo.

Se tiene el arreglo con el número que se va a iterar

1	2	3	5	4	2
---	---	---	---	---	---

Se hace un recorrido sobre el arreglo

1	2	3	5	4	2
---	---	---	---	---	---

→→→→→

Se encuentran las tuplas de números que se deben cambiar para lograr la condición de obtener el siguiente número mayor utilizando los mismos números

***esta operación se hace tantas veces como iteraciones sean indicadas por el parámetro, este caso 2 iteraciones**

1.

1	2	3	5	4	2	→	1	2	4	2	3	5
---	---	---	---	---	---	---	---	---	---	---	---	---

2.

1	2	4	2	3	5	→	1	2	4	2	5	3
---	---	---	---	---	---	---	---	---	---	---	---	---

De esta forma se obtiene

1	2	3	5	4	2	→	1	2	4	2	5	3
---	---	---	---	---	---	---	---	---	---	---	---	---

Ya hablando desde el algoritmo. Primero se hace la lectura de los datos. A partir de ellos se utiliza el método minPerMaxAnterior que recibe como parámetros un arreglo de caracteres de números y retorna un número mayor al anterior con los mismos números que lo componen pero menor a cualquier siguiente iteración posible. Para esto lo que se hace es que se busca un número mayor del lado derecho y un número menor del lado derecho que al intercambiarlos genere el resultado adecuado. En cada iteración se pueden realizar uno o más cambios de números.

Por otro lado, el invariante vienen a ser los número cuya posición no puede ser cambiada desde un comienzo pues causarían que el número resultante fuera menor al número con que se empieza.

2. Análisis de complejidades espacial y temporal.

Análisis temporal:

La complejidad temporal $T(k,n) = O(kn)$ ya que el método minPerMaxAnterior en términos generales se demora n y este se debe realizar k veces para sacar el numero de la iteración k mayor a los k anteriores que contengan los mismos números. Debe tenerse en cuenta que este k tiene un maximo ya que cuando no hay siguiente iteración se retorna $*$ y se hace terminacion del programa.

Análisis espacial:

Como sólo se usan variables auxiliares como contadores o para guardar las cadenas necesarias, la complejidad espacial del algoritmo es $S(k,n) = O(n)$ ya que se tiene en cuenta el espacio que se necesita para gradar los caracteres del numero dado por parametro.

3. Comentarios finales.

En los escenarios de prueba proporcionados en el enunciado se logra obtener los resultados esperados. Además se implementaron nuevos casos de prueba donde logró dar los resultados correctos y además de esto obtuvo un buen desempeño.