????? Tercera parte

Implementar un algoritmo que resuelva el siguiente problema:

$P \equiv \{0 < n < 10.000\}$
proc *xxxx* (**int** $a[]$, **int** n, **out int** $p[]$, **out int** t)
$Q \equiv \{(0 \leq t < n) \wedge (\forall x : 0 \leq x < t : menores(a, n, p[x])) \wedge (\forall k : 0 \leq k < n \wedge menores(a, n, k) : pertenece(k, p))\}$

donde:

- $menores(a, n, p) \equiv \forall u, w : 0 \leq u \leq p < w < n : a[u] < a[w]$ y
- $pertenece(k, p) \equiv \exists s : 0 \leq s < t : k = p[s]$.

Requisitos de implementación.
El orden de complejidad del algoritmo debe ser lineal respecto al número de elementos del vector. Es suficiente con recorrer el vector dos veces.
Hint.
Se puede utilizar memoria auxiliar para almacenar información sobre el vector y evitar el coste cuadrático.

Entrada

La entrada comienza con un valor entero que indica el número de casos de prueba. Cada caso de prueba consta de dos líneas. La primera contiene el valor de n. La segunda línea contiene los valores del vector.

Salida

Para cada caso de prueba se escribe en una línea los valores de p separados por un blanco cada uno. No debe escribirse un blanco al final de la línea.

Entrada de ejemplo

```
4
10
3 4 2 9 7 6 8 17 20 22
15
2 1 3 1 2 6 5 6 9 10 10 9 13 15 13
4
4 3 2 1
5
1 2 3 4 5
```

Salida de ejemplo

```
2 6 7 8
4 7 11

0 1 2 3
```

Autor: Isabel Pita.