Estructuras de Datos y Algoritmos

Grados en Ingeniería Informática

Examen Primer Cuatrimestre, 10 de Febrero de 2017.

Nombre: _____ **Grupo:** _____

Laboratorio: _____ **Puesto:** _____

1. (4 puntos) Dado un vector de $n \geq 0$ enteros, se pide diseñar un algoritmo eficiente que (sin utilizar un array auxiliar) modifique el vector de forma que todos los elementos mayores o iguales que 0 queden colocados al principio del vector y los estrictamente menores que 0 a continuación, y que adicionalmente devuelva cual es la posición de separación entre positivos y negativos, siendo esta la posición del primer número negativo en el vector modificado o n en caso de que no haya ninguno. Se pide:

1. (1 punto) Especificar el problema.
2. (1,5 puntos) Diseñar e implementar un algoritmo que resuelva el problema.
3. (1 punto)Escribir un invariante y una función de cota que permitan demostrar la corrección del algoritmo implementado.
4. (0,5 puntos) Justificar el coste del algoritmo.

Entrada

La primera línea contiene un número que indica el número de casos de prueba que aparecen a continuación. Cada caso de prueba se compone de dos líneas. La primera de ellas tiene el número de elementos del vector. La segunda contiene los elementos del vector separados por blancos.

Salida

Para cada caso de prueba se escribirá en una línea el vector modificado y en otra línea la posición de separación. Se ha de tener en cuenta que la salida mostrada en el ejemplo es solamente una de las posibles.

Entrada de ejemplo

```
7
1
2
1
-3
6
5 4 1 9 0 2
5
-3 -1 -2 -7 -8
7
1 -3 2 -1 9 -6 -10
10
3 7 -100 1 0 1 4 6 8 100
10
0 -3 -4 -1 -9 -6 0 -5 -10 -20
```

Salida de ejemplo

```
2
1
-3
0
5 4 1 9 0 2
6
-3 -1 -2 -7 -8
0
1 9 2 -1 -3 -6 -10
3
3 7 100 1 0 1 4 6 8 -100
9
0 0 -4 -1 -9 -6 -3 -5 -10 -20
2
```

2. (3 puntos) Supongamos dado un vector ordenado de $n \geq 1$ elementos, en el que todos los elementos aparecen repetidos dos veces, excepto uno que solamente aparece una vez (por tanto n es impar). Se pide diseñar un algoritmo eficiente que devuelva la posición de dicho elemento. Justifica el coste del algoritmo, para lo cual has de plantear la recurrencia correspondiente.

Entrada

La primera línea contiene un número que indica el número de casos de prueba que aparecen a continuación. Cada caso de prueba se compone de dos líneas. La primera de ellas contiene el número de elementos del vector. La segunda contiene los elementos del vector separados por blancos.

Salida

Para cada caso de prueba se escribirá en una línea la posición del elemento que aparece una única vez.

Entrada de ejemplo

```
9
1
3
3
1 2 2
5
1 1 2 9 9
5
1 2 2 9 9
5
1 1 2 2 9
7
3 5 5 9 9 11 11
7
3 3 5 9 9 11 11
7
3 3 5 5 9 11 11
7
3 3 5 5 9 9 11
```

Salida de ejemplo

```
0
0
2
0
4
0
2
4
6
```

3. (3 puntos) Se desea rellenar un vector de 2n posiciones (siendo $n \geq 1$) con dos apariciones de cada uno de los números 1 a n, de tal forma que el número de posiciones entre dos apariciones (excluyendo a estas) de cada número i es igual a i. Implementar un algoritmo que muestre todas las soluciones posibles y cuantas son.

Entrada

La primera línea contiene un número que indica el número de casos de prueba que aparecen a continuación. Cada caso de prueba se compone de una línea en la que aparece el entero n.

Salida

Para cada caso de prueba se mostrarán las posibles soluciones, cada una en una línea (no necesariamente en el orden que se muestra en el ejemplo) y al finalizar una linea indicando el número de soluciones encontradas, según el formato del ejemplo.

Entrada de ejemplo

```
7
1
2
3
4
5
6
7
```

Salida de ejemplo

```
Soluciones: 0
Soluciones: 0
2 3 1 2 1 3
3 1 2 1 3 2
Soluciones: 2
2 3 4 2 1 3 1 4
4 1 3 1 2 4 3 2
Soluciones: 2
Soluciones: 0
Soluciones: 0
1 4 1 5 6 7 4 2 3 5 2 6 3 7
1 4 1 6 7 3 4 5 2 3 6 2 7 5
1 5 1 4 6 7 3 5 4 2 3 6 2 7
1 5 1 6 3 7 4 5 3 2 6 4 2 7
1 5 1 6 7 2 4 5 2 3 6 4 7 3
1 5 1 7 3 4 6 5 3 2 4 7 2 6
1 6 1 3 5 7 4 3 6 2 5 4 2 7
1 6 1 7 2 4 5 2 6 3 4 7 5 3
1 7 1 2 5 6 2 3 4 7 5 3 6 4
1 7 1 2 6 4 2 5 3 7 4 6 3 5
2 3 6 2 7 3 4 5 1 6 1 4 7 5
2 3 7 2 6 3 5 1 4 1 7 6 5 4
. . .
Soluciones: 52
```