14.5 2010

Pregledna plošča / Moji predmeti / aps2uni / 25. marec - 31. marec / Izziv 4

Začeto dne	torek, 26. marec 2019, 10:08
Stanje	Zaključeno
Dokončano dne	ponedeljek, 1. april 2019, 00:00
Porabljeni čas	5 dni 12 ure
Točke	1,00/1,00
Ocena	<b>10.00</b> od možne ocene 10.00 ( <b>100</b> %)

1-----

Vprašanje **1**Pravilno
Ocena 1,00 od
1.00

Napišite program, ki tabelo 32 bitnih celih števil uredi glede na število bitov, ki so enaki 1 v dvojiški predstavitvi tega števila

Npr. tabela 6 5 4 1 3 bi bila urejena 4 1 6 5 3, ker 4 = 100, 1 = 1, 6 = 110, 5 = 101 in 3 = 11. Števila, ki imajo enako število enic ostanejo v istem vrstnem redu kot pred urejanjem.

Za urejanje uporabite stabilno urejanje s štetjem (counting sort). Na standardnem vhodu boste najprej prejeli dolžino tabele n, za tem pa n celih števil.

Na standardni izhod izpisujete potek vpisovanja elementov v končno (urejeno tabelo). Za vsak zapis zapišete v eni vrstici par (el, pos), kjer je el število, ki ga zapisujete, pos pa indeks v tabeli kamor ta element zapisujete. V zadnji vrstici izpišite še urejeno tabelo ločeno s presledki.

## For example:

Input	Result
5	(3,4)
6 5 4 1 3	(1,1)
	(4,0)
	(5,3)
	(6,2)
	4 1 6 5 3

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
2
3 ,
   public class Izziv4 {
4
        public static void main(String [] args) {
           Scanner sc = new Scanner (System.in);
6
7
            int dolzina = sc.nextInt();
           int tabela [] = new int [dolzina];
8
9
10
            for (int i = 0; i < tabela.length; i++)</pre>
                tabela[i] = sc.nextInt();
11
12
13
            int [] urejena = uredi(tabela, dolzina);
14
15
            for (int i = 0; i < dolzina; i++)</pre>
               System.out.print(urejena[i] + " ");
16
17
18
        private static int [] uredi (int [] tabela, int dolzina) {
19
20
            int [] urejena = new int [dolzina];
            21
```

	Input	Expected	Got	
~	8	(1,3)	(1,3)	~
	8 7 6 5 4 3 2 1	(2,2)	(2,2)	
		(3,6)	(3,6)	
		(4,1)	(4,1)	
		(5,5)	(5,5)	
		(6,4)	(6,4)	
		(7,7)	(7,7)	
		(8,0)	(8,0)	
		8 4 2 1 6 5 3 7	8 4 2 1 6 5 3 7	
~	10	(636,7)	(636,7)	~
	820 37 908 764 797 460 641	(704,2)	(704,2)	
	719 704 636	(719,9)	(719,9)	
		(641,1)	(641,1)	
		(460,5)	(460,5)	
		(797,6)	(797,6)	
		(764,8)	(764,8)	
		(908,4)	(908,4)	
		(37,0)	(37,0)	
		(820,3)	(820,3)	
		37 641 704 820 908 460 797	37 641 704 820 908 460 797	
		636 764 719	636 764 719	

	I				
	Input	Expected	Got		
~	15	(405,8)	(405,8)	~	
	703 224 248 924 122 727 495	(568,2)	(568,2)		
	440 462 89 107 879 626 568	(626,7)	(626,7)		
	405	(879,14)	(879,14)		
		(107,6)	(107,6)		
		(89,1)	(89,1)		
		(462,10)	(462,10)		
		(440,5)	(440,5)		
		(495,13)	(495,13)		
		(727,11)	(727,11)		
		(122,4)	(122,4)		
		(924,9)	(924,9)		
		(248,3)	(248,3)		
		(224,0)	(224,0)		
		(703,12)	(703,12)		
		224 89 568 248 122 440 107	224 89 568 248 122 440 107		
		626 405 924 462 727 703 495	626 405 924 462 727 703 495		
		879	879		
~	20	(613,8)	(613,8)	~	
	381 435 567 206 342 217 175	(703,19)	(703,19)		
	508 18 158 374 773 711 685	(732,15)	(732,15)		
	358 516 687 732 703 613	(687,18)	(687,18)		
		(516,1)	(516,1)		
		(358,7)	(358,7)		
		(685,14)	(685,14)		
		(711,13)	(711,13)		
		(773,2)	(773,2)		
		(374,12)	(374,12)		
		(158,6)	(158,6)		
		(18,0)	(18,0)		
		(508,17)	(508,17)		
		(175,11)	(175,11)		
		(217,5)	(217,5)		
		(342,4)	(342,4)		
		(206,3)	(206,3)		
		(567,10)	(567,10)		
		(435,9)	(435,9)		
		(381,16)	(381,16)		
		18 516 773 206 342 217 158	18 516 773 206 342 217 158		
		358 613 435 567 175 374 711	358 613 435 567 175 374 711		
		685 732 381 508 687 703	685 732 381 508 687 703		

3/5

	Immir. 4			
	Input	Expected	Got	
~	30 231 725 216 534 174 77 783 969 820 234 299 812 70 949 800 125 891 916 229 908 60 138 911 989 973 144 506 542 278 932	(932,18) (278,8) (542,17) (506,27) (144,0) (973,26) (989,29) (911,25) (138,3) (60,7) (908,16) (229,15) (916,14)	(932,18) (278,8) (542,17) (506,27) (144,0) (973,26) (989,29) (911,25) (138,3) (60,7) (908,16) (229,15) (916,14)	~
		(891,28) (125,23) (800,2) (949,24) (70,1) (812,13) (299,12) (234,11) (820,10) (969,22) (783,21) (77,6) (174,9)	(891,28) (125,23) (800,2) (949,24) (70,1) (812,13) (299,12) (234,11) (820,10) (969,22) (783,21) (77,6) (174,9)	
		(534,5) (216,4) (725,20) (231,19) 144 70 800 138 216 534 77 60 278 174 820 234 299 812 916 229 908 542 932 231 725 783 969 125 949 911 973 506 891 989	(534,5) (216,4) (725,20) (231,19) 144 70 800 138 216 534 77 60 278 174 820 234 299 812 916 229 908 542 932 231 725 783 969 125 949 911 973 506 891 989	
~	5 6 5 4 1 3	(3,4) (1,1) (4,0) (5,3) (6,2) 4 1 6 5 3	(3,4) (1,1) (4,0) (5,3) (6,2) 4 1 6 5 3	~

Passed all tests! 🗸

Question author's solution:

14.5 2010

```
import java.util.Scanner;
public class Izziv4 {
  private static int[] countingSort(int[] a){
          int[] result = new int[a.length];
          int[] counts = new int[33];
          for (int i : a)
                counts[Integer.bitCount(i)]++;
          int cumSum = 0;
          for (int i=0; i<33 ; i++ ) {
                cumSum += counts[i];
                counts[i] = cumSum-1;
          for (int i = a.length-1; i>= 0 ; i--) {
                 int pos = Integer.bitCount(a[i]);
                  result[counts[pos]] = a[i];
                  System.out.println("("+a[i]+","+counts[pos]+")");
                  counts[pos]--;
          return result;
  }
  public static void main(String[] args) {
   Scanner sc = new Scanner(System.in);
   int n = sc.nextInt();
       int[] a = new int[n];
       for (int i = 0; i < n; i++)
                a[i] = sc.nextInt();
       a = countingSort(a);
       for (int i : a )
               System.out.print(i+" ");
  }
```

Pravilno

Marks for this submission: 1,00/1,00.

■ Kviz 4

Skok na...

5 - Deli in vladaj ►