Wykład 1_3

Algorytmy sortowania liniowego - sortowanie przez zliczanie

Zadanie: Należy posortować *N* elementów. Każdy element jest liczbą całkowitą z przedziału od 1 do *K*.

Algorytm sortowania N elementów - poziom konceptualny:
Sortowanie_przez_zliczanie:
(1) *wskaż na pierwszy licznik;*
(2) *wykonaj, co następuje,* ***K*** *razy:*
  (2.1) *wyzeruj wskazany licznik;*
  (2.2) *wskaż na następny licznik;*

{Wyznaczenie liczby elementów o danej wartości od 1 do **K**}
(3) *wskaż na pierwszy element w ciągu wejściowym;*
(4) *wykonaj, co następuje,* ***N*** *razy:*
  (4.1) *dodaj 1 do zawartości licznika zliczającego wystąpienie danej wartości, przechowywanej we wskazanym elemencie;*
(4.2) *wskaż na następny element;*

{Wyznaczenie, ile elementów jest mniejszych i równych danej wartości. Określa ona pozycję tej wartości w ciągu posortowanym.}
(5) *wskaż na licznik drugiej wartości w dziedzinie wartości sortowanych;*
(6) *wykonaj, co następuje,* ***K-1*** *razy:*
  (6.1) *dodaj do zawartości wskazanego licznika zawartość licznika poprzedniej wartości;*
(6.2) *wskaż na następny licznik;*

{Ustawienie elementów w ciągu wyjściowym w porządku rosnącym}
(7) *wskaż na ostatni element w ciągu wejściowym;*
(8) *wykonaj, co następuje,* ***N*** *razy:*
  (8.1) *wskaż na licznik wartości odpowiadającej zawartości wskazanego elementu;*
  (8.2) *umieść wskazany element z ciągu wejściowego w ciągu wyjściowym na pozycji określonej zawartością licznika;*
  (8.3) *zmniejsz zawartość licznika o 1;*
  (8.4) *wskaż na następny element w ciągu wejściowym.*

Kroki	Ciąg wejściowy								Liczniki						Ciąg wyjściowy							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*1*	*2*	*3*	*4*	*5*	*6*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
2	3	6	4	1	3	4	1	4	0	0	0	0	0	0								
4									2	0	2	3	0	1								
6									2	2	4	7	7	8								
7								4														
1/8.1								4				**7**										
8.2												**7**									**4**	
8.3-4							1		2	2	4	**6**	7	8							**4**	
2/8.1							1		**2**												4	
8.2									**2**							**1**					4	
8.3-4						4			**1**	2	4	6	7	8		**1**					4	
3/8.1						4						**6**				1					4	
8.2												**6**				1				**4**	4	
8.3-4					3				1	2	4	**5**	7	8		1				**4**	4	
4/8.1					3						**4**					1				4	4	
8.2											**4**					1		**3**		4	4	
8.3-4				1					1	2	**3**	5	7	8		1		**3**		4	4	
5/8.1				1					**1**							1		3		4	4	
8.2									**1**						**1**	1		3		4	4	
8.3-4			4						**0**	2	3	5	7	8	**1**	1		3		4	4	
6/8.1			4									**5**			1	1		3		4	4	
8.2												**5**			1	1		3	**4**	4	4	
8.3-4		6							0	2	3	**4**	7	8	1	1		3	**4**	4	4	
7/8.1		6												**8**	1	1		3	4	4	4	
8.2														**8**	1	1		3	4	4	4	**6**
8.3-4	3								0	2	3	4	7	**7**	1	1		3	4	4	4	**6**
8/8.1	3										**3**				1	1		3	4	4	4	6
8.2											**3**				1	1	**3**	3	4	4	4	6
8.3-4									0	2	**2**	4	7	7	1	1	**3**	3	4	4	4	6

Tab2. Kolejne kroki algorytmu sortowania przez zliczanie

Algorytm sortowania przez zliczanie - poziom projektowy

(1) *i ← 1*
(2) *dopóki i <= K, wykonuj co następuje:*
{zerowanie liczników , czyli elementów w tablicy *Liczniki*}
(2.1) *Liczniki(i) ← 0;*
(2.2) *i ← i+1;*
(3) *j ← 1;*
(4) *dopóki j <= N, wykonuj, co następuje:*
{zliczanie elementów z tablicy *We* o wartości *i* w tablicy *Liczniki*}
(4.1) *i ← We(j);*
(4.2) *Liczniki(i) ← Liczniki(i) + 1;*
(4.3) *j ← j+1*
(5) *i ← 2;*
(6) *dopóki i <= K, wykonuj, co następuje:*
{zliczanie elementów mniejszych / równych *i* umieszczonych w tablicy *We*}
(6.1) *Liczniki (i) = Liczniki(i) + Liczniki(i-1);*
(6.2) *i ← i +1;*
(7) *i ← N;*
(8) *dopóki i >= 1, wykonuj, co następuje:*
{Ustawianie elementów w tablicy *Wy* w porządku niemalejącym, ustalając ich indeksy zgodnie z zawartością elementów tablicy *Liczniki*}
(8.1) *j ← We(i);*
(8.2) *Wy(Liczniki (j)) ← We(i);*
(8.3) *Liczniki(j) ← Liczniki(j) -1;*
(8.4) *i ← i - 1;*

Cechy algorytmu:

1. Zależność liniowa czasu wykonania od liczby elementów.
2. Stabilność: elementy o tych samych wartościach występują w tablicy wynikowej w takiej samej kolejności jak w tablicy początkowej

```
//---------------------------------------------------------------------------
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <Math>
#pragma hdrstop
//---------------------------------------------------------------------------
#pragma argsused
const long N=8;
const int K=10;

void wyswietl(int t[], long ile);
void zliczanie(int We[], int Wy[], long l, long p);

void main()
{  int We[N]={3,6,4,1,3,4,1,4};
   int Wy[N];
   long ile=N;

   clrscr();
   zliczanie(We, Wy, 0L, ile-1);
   wyswietl(Wy, ile);
   getch();
}
void wyswietl(int t[], long ile)
 { for (long i=0L; i<ile; i++)
          printf("%i \n", t[i]); }

  void zliczanie(int We[], int Wy[], long l, long p)
   { int Cyfry[K], j;
                        //zerowanie liczników
     for (int i= 0; i< K; i++)          Cyfry[i] = 0;
                        //zliczanie elementów z tablicy We w tablicy Cyfry
     for (long i = l; i<= p; i++)      Cyfry[We[i]]+= 1;
                        //zliczanie elementów mniejszych oraz równych "i" w tablicy Cyfry
     for (long i=1; i<K;  i++)         Cyfry[i] += Cyfry[i-1];
     for (long i=p; i>=l; i--)
 /*sortowanie elementów z tablicy We: element We[i] jest indeksem elementu tablicy Cyfry, zawierającego indeks elementu w tablicy Wy, w którym
jest wstawiony element We[i]*/
         {    j = We[i];
              Wy[Cyfry[j]-1] = We[i];
              Cyfry[j] -= 1;
         }
  }
```

Analiza algorytmu sortowania przez zliczanie

Lp		Koszt	Liczba wykonań
1	**for** (**int** i= 0; i< K; i++)	c_1	*K+1*
2	Cyfry[i] = 0;	c_2	*K*
3	**for** (**long** i = 0; i<= N-1; i++)	c_3	*N+1*
4	Cyfry[We[i]]+= 1;	c_4	*N*
5	**for** (**long** i=1; i<K; i++)	c_5	*K*
6	Cyfry[i] += Cyfry[i-1];	c_6	*K-1*
7	**for** (**long** i=N-1; i>=0; i--)	c_7	*N+1*
8	{ j = We[i];	c_8	*N*
9	Wy[Cyfry[j]-1] = We[i];	c_9	*N*
10	Cyfry[j] -= 1; } }	c_{10}	*N*

$$T(N,K) = (K+1)\,c_1 + K\,c_2 + (N+1)\,c_3 + N\,c_4 + K\,c_5 + (K-1)\,c_6 + (N+1)\,c_7 + N\,c_8 + N\,c_9 + N\,c_{10} = K\,(c_1 + c_2 + c_5 + c_6) + N(c_3 + c_4 + c_7 + c_8 + c_9 + c_{10}) + (c_1 + c_3 - c_6 + c_7)$$

$$T(N, K) = c' K + c'' N + c''' \approx c'' N + c \qquad (\text{dla } K = 10)$$

Wniosek: Czas wykonania algorytmu sortowania przez zliczanie jest liniowo zależny od liczby danych wejściowych.