Proiectarea algoritmilor

Lucrare de laborator nr. 4

Paradigma Divide_et_Impera - Algoritmul lui Batcher de sortare bitonică

Cuprins

1	Algoritmul lui Batcher de sortare bitonică		
	1.1	Descriere	1
	1.2	Pseudocod	2
	1.3	Exemplu de sortare a unei secvențe bitone	2
2	Sarc	zini de lucru	3

1 Algoritmul lui Batcher de sortare bitonică

1.1 Descriere

- Notatii:
 - a [0..n-1] este un tablou unidimensional de dimensiune n.
 - (a, i, d) definește segmentul a[i]..a[i+d-1] din tabloul a.
 - s este un parametru binar care specifică ordinea crescătoare (s = 0) sau descrescătoare
 (s = 1) a cheilor de sortare.
 - compara_si_schimba(x, y, s) desemnează sortarea a două elemente x și y ordinea indicată de parametrul s.
- Premise: Inițial, a [0..n-1] conține secvența de sortat.
- Apel: BatcherSort(a, 0, n, 0) sau BatcherSort(a, 0, n, 1).

1.2 Pseudocod

```
procedure BatcherSort(a, i, d, s)
   if (d = 2)
      then
            (a[i], a[i+1]) \leftarrow compara_si_schimba(a[i], a[i+1], s)
      else
            BatcherSort(a, i, d/2, 0)
            BatcherSort(a, i + d/2, d/2, 1)
            sortare_secventa_bitona(a, i, d, s)
end
procedure sortare_secventa_bitona(a, i, d, s)
   if (d = 2)
      then
            (a[i], a[i+1]) \leftarrow compara_si_schimba(a[i], a[i+1], s)
      else
            for j \leftarrow 0 to d/2-1 do
                 (a[i+j], a[i+j+d/2]) \leftarrow
                       compara_si_schimba(a[i+j], a[i+j+d/2], s)
            sortare_secventa_bitona(a, i, d/2, s)
            sortare\_secventa\_bitona(a, i + d/2, d/2, s)
end
```

1.3 Exemplu de sortare a unei secvențe bitone

Original

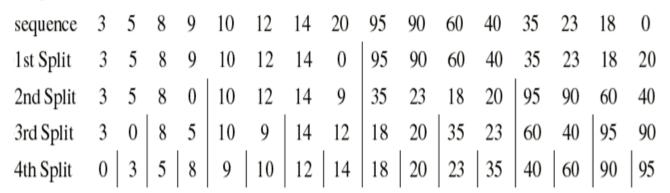


Figura 1: Copyright @ 1994 Benjamin/Cummings Publishing Co.

2 Sarcini de lucru

- 1. Scrieți o funcție C/C++ care implementează algoritmul BatcherSort;
- 2. Măsurați timpii de execuție pentru n elemente sortate ($10.000 \le n \le 10.000.000$) considerând trei cazuri diferite: elementele sunt deja sortate crescător și le sortăm tot crescător, elementele sunt sortate descrescător și le sortăm crescător iar în final elementele sunt distribuite aleator și le sortăm crescător. Interpretați rezultatele.

Bibliografie

[1] Lucanu, D. și Craus, M., Proiectarea algoritmilor, Editura Polirom, 2008.