

Temă 5

Ex1: Să se determine eficiența în termeni de $O(f(n))$ pentru sortarea prin inserție în cazul în care toate elementele din setul de date de intrare au aceeași valoare.

Ex2: Modificați algoritmul pentru sortarea prin inserție astfel încât sortarea să se facă de la sfârșitul tabloului spre început. Ordinea va rămâne tot crescătoare.

Ex3: Scrieți un program bazat pe algoritmul de sortare prin inserție, care să primească ca input o stivă (și nu un tablou unidimensional, ca în exemplul din curs). Algoritmul poate folosi un număr finit de stive și de variabile auxiliare (nu poate folosi tablouri) și trebuie să fie $O(n^2)$ în cazul cel mai defavorabil.

1)

```
void insertion_sort(tip_element a[], int n)
{
    int i, j;
    tip_element tmp;
    for (i = 1; i < n; i++)
    {
        tmp = a[i]; /*salvarea elementului de inserat, intr-o
zona de memorie tampon*/
        for (j = i; (j>0) && (tmp.cheie < a[j-1].cheie); j--)
            a[j] = a[j - 1];
        /* mutam elementele din sirul sortat cu cheia mai
mare decat a elementului de inserat spre dreapta */
        a[j] = tmp; /* inseram elementul la locul potrivit
in sirul sortat */
    }
}
```

Daca toate elementele sunt egale, NU se vor face schimbari, dar un nr minim de comparatii tot vor avea loc.

Numărul total de comparații de chei(C):

• Cazul cel mai favorabil:

$$C_{min} = \sum_{i=1}^{n-1} 1 = n - 1 = O(n)$$

• Cazul cel mai defavorabil:

$$C_{max} = \sum_{i=1}^{n-1} i = 1 + 2 + \dots + (n-1) = \frac{n(n-1)}{2} = O(n^2)$$

• Cazul mediu:

$$C_{me} = \sum_{i=1}^{n-1} \frac{i+1}{2} = \sum_{i=1}^{n-1} \frac{i}{2} + \sum_{i=1}^{n-1} \frac{1}{2} = \frac{1}{2} \frac{n(n-1)}{4} + \frac{1}{2}(n-1) = \frac{n^2 + n - 2}{4} = O(n^2)$$

Ne aflam, deci, în cazul cel mai favorabil și vom avea complexitatea $O(n)$.