Lucrarea de Laborator nr. 8 (Opțional)
Tema: Analiza empirică a algoritmilor de sortare și de căutare

Scopul: Studierea posibilităților și mijloacelor limbajului C de programare a algoritmilor de sortare și de căutare pentru tablouri unidimensionale și obținerea deprinderilor de analiză empirică a algoritmilor

Sarcina

Să se scrie un program în limbajul C pentru analiza empirică a algoritmului propus (după variantă) cu crearea funcției de căutare sau sortare pentru tabloul unidimensional de *n* elemente cu afișarea la ecran a următorului meniu de opțiuni:

1. Tabloul demonstrativ de *n* elemente (5<= n <=20).
2. Tabloul cu valori aleatorii (n=10000, n=100000, n=1000000).
3. Tabloul sortat crescător (n=10000, n=100000, n=1000000).
4. Tabloul sortat descrescător (n=10000, n=100000, n=1000000).
5. Ieșire din program.

Analiza empirică a algoritmului constă în: a) determinarea timpului de rulare, numărului de comparații, numărului de schimbări (sau deplasări) de elemente pentru tablouri cu 3 diferite seturi de valori și cu 3 diferite volume de elemente; b) compararea și analiza rezultatelor obținute utilizând funcția creată și funcția din biblioteca standard a limbajului C; c) tragerea concluziilor.

Variante de algoritmi:

1. Căutare secvențială.
2. Căutare binară cu algoritmul iterativ.
3. Sortare bulelor în ordine ascendentă.
4. Sortare prin selecție liniară în ordine descendentă.
5. Sortare prin selecție și schimb în ordine ascendentă.
6. Sortare prin inserție în ordine descendentă.
7. Sortare shaker în ordine ascendentă.
8. Căutare binară cu algoritmul recursiv.
9. Sortare Shell în ordine ascendentă.
10. Sortare rapidă în ordine descendentă.
11. Sortare prin interclasare (merge sort).
12. Sortare qsort ().
13. Căutare bsearch ().
14. Sortare prin coada de prioritate.
15. Sortare heap în ordine descendentă.
16. Sortare utilizând arbore binar de căutare.
17. Sortare prin interclasare (merge sort) în ordine ascendentă.
18. Sortare Shell în ordine descendentă.
19. Sortare prin inserție în ordine ascendentă.
20. Sortare heap în ordine descendentă.