Metode de căutare

Căutare secvențială

Căutarea secvențială este unul dintre cei mai simpli algoritmi studiați. El urmăreşte să verifice apartenența unui element la un şir de elemente de aceeaşi natură, în speță a unui număr la un şir de numere. Pentru aceasta se parcurge şirul de la un capăt la celălalt şi se compară numărul de căutat cu fiecare număr din şir. În cazul în care s-a găsit corespondență (egalitate), un indicator flag este poziționat. La sfârşitul parcurgerii şirului, indicatorul ne va arăta dacă numărul căutat aparține sau nu şirului.

```
#include<iostream>
using namespace std;
int main()
{
	int n,x,i,gasit,a[50];
	cout<<"Dati numarul de elemente ale sirului : "; cin>>n;
	cout<<"Dati numarul de cautat : "; cin>>x;
	cout<<"Dati elementele sirului"<<endl;
	gasit=0;
	for(i=1; i<=n; i++){
		cout<<"a["<<i<<"]= "; cin>>a[i];
		if (a[i]==x) gasit=1;
	}
	if (gasit==0)
		cout<<"Numarul nu apartine sirului
	!"<<endl; else cout<<"Numarul apartine
	sirului !"<<endl;
}
```

Temă. Să se transforme programul anterior astfel încât să se afişeze pozițiile în care apare elementul căutat.

Căutare binară

Algoritmul de căutare binară oferă performanțe mai bune decât algoritmul de căutare secvențială. El funcționează astfel: se compară numărul de căutat cu elementul aflat la mijlocul şirului (element care se mai numeşte şi pivot). În cazul în care cele două elemente coincid căutarea s-a încheiat cu succes. Dacă numărul de căutat este mai mare decât pivotul, se continuă căutarea î n aceeaşi manieră în subşirul delimitat de pivot şi capătul şirului inițial. Dacă numărul de căutat este mai mic decât pivotul se continuă căutarea în aceeaşi manieră în subşirul delimitat de pivot şi începutul şirului inițial.

Algoritmul prezentat se încadrează în clasa algoritmilor elaborați conform tehnicii de programare *Divide et Impera*.

Unul din dezavantajele acestui algoritm este că şirul în care se face căutarea trebuie să fie inițial sortat.

```
#include<iostream>
using namespace std;
int main()
{
	int i, n, x, st, dr, mijl, gasit, a[50];
	cout<<"Introduceti numarul elementelor vectorului : "; cin>>n;
	cout<<"Introduceti elementul de cautat : ";
	cin>>x;
	for(i=1;i<=n;i++) {
		cout<<"a["<<i<<"]= ";
		cin>>a[i];
			}
	st=1; dr=n; gasit=0;
	while (st<=dr && gasit!=1)
	{
		mijl=(st+dr)/2;
		if (a[mijl]==x)gasit=1;
		else
			if (x<a[mijl])dr=mijl-1;
			else
				st=mijl+1;
	}
	if (st>dr) cout<<"Nu s-a gasit elementul cautat !"<<endl;
	else cout<<"Elementul cautat apartine sirului !"<<endl;
}
```