UNIVERSITATEA DE STAT DIN MOLDOVA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALITATEA INFORMATICA

RAPORT

la disciplina „Algoritme, Structuri de Date și Complexitate”

Lucrarea de laborator nr. 4: Metode de acces la elementele unui masiv

Conducătorul științific: Novac Ludmila, dr. conf. univ.
Autor: Cemîrtan Cristian, student din grupa I2101

CHIȘINĂU – 2023

Cuprins

Preliminări

În cadrul lucrării de laborator, vectorul B care va fi testat este în felul următor: [-3..1][-5..4][100..103][0..1][-1..0][-10..-9][99..101], care este 7-dimensional. Se menționează că programul elaborat în limbajul C suportă universal orice dimensiune pentru un vector.

Pentru a măsura consumul de memorie utilizat de fiecare tip dinamic de date, am implementat propria mea utilitate de gestionare a memoriei. Această utilitate contorizează numărul alocărilor de memorie într-o sesiune de lucru.

De asemenea, fișierul textual utilizat în cadrul lucrării de laborator a fost preluat din lucrarea precedentă.

Testarea metodelor de acces

Metoda directă.

Este cea mai simplă metodă, și nu lasă adițional nici o amprentă de memorie, indiferent cum este reprezentat vectorul în memoria operativă.

```
>mem
Amprenta de memorie: 0 alocari (0 bytes)
Marimea tabloului: 4800 elemente
```

Figura 1

Metoda accelerată cu ajutorul vectorului definitoriu.

Prin metoda accelerată se are în vedere utilizarea valorilor salvate în loc de a le recalcula la fiecare accesare unui element din vector. Valorile salvate pot fi definite ca: suma lj * Dj și mărimile Dj, unde j este între 0 și k – 1 inclusiv. În cazul nostru, k = 7 și vom lăsa amprenta de memorie: 1 * sizeof(size_t) + 7 * sizeof(size_t) = 64 octeți. Amprenta de memorie rămâne același indiferent de reprezentarea vectorului în memoria operativă.

```
>mem
Amprenta de memorie: 1 alocari (64 bytes)
Marimea tabloului: 4800 elemente
```

Figura 2

Vectorul Iliffe.

Spre deosebire de metodele menționate anterior, această metodă implică alocarea vectorilor de pointeri, pentru e elimina necesitatea operațiilor de înmulțire. Aici, amprenta de memorie se diferă după reprezentarea vectorului. Analizând figurile 3 și 4, putem spune că reprezentarea vectorului pe coloane este mai eficientă în privința consumului de memorie operativă.

Știind că mărimea tabloului este: (1 - -3 + 1) * (4 - -5 + 1) * (103 – 100 + 1) * (1 – 0 + 1) * (0 - -1 + 1) * (-9 - -10 + 1) * (101 – 99 + 1) = 5 * 10 * 4 * 2 * 2 * 2 * 3 = 4800:

- Vectorul Iliffe pe linii va avea în total 5 + (5 * 10) + (5 * 10 * 4) + (5 * 10 * 4 * 2) + (5 * 10 * 4 * 2 * 2) + (5 * 10 * 4 * 2 * 2 * 2) = 5 + 50 + 200 + 400 + 800 + 1600 = 3055 pointeri (24440 octeți);
- Vectorul Iliffe pe coloane va avea 3 + (3 * 2) + (3 * 2 * 2) + (3 * 2 * 2 * 2) + (3 * 2 * 2 * 2 * 4) + (3 * 2 * 2 * 2 * 4 * 10) = 3 + 6 + 12 + 24 + 96 + 960 = 1101 pointeri (8808 octeți).

```
>mem
Amprenta de memorie: 1456 alocari (24440 bytes)
Marimea tabloului: 4800 elemente
```

Figura 3. Pe linii

```
>mem
Amprenta de memorie: 142 alocari (8808 bytes)
Marimea tabloului: 4800 elemente
```

Figura 4. Pe coloane

Compararea metodelor de acces

Pentru tabloul [-3..1][-5..4][100..103][0..1][-1..0][-10..-9][99..101], se repetat accesarea elementelor de 1000 ori:

Tabela 1

Metoda	Reprezentare	Acces aliator (ms)	Acces liniar (ms)
Directă	Pe linii	**3211**	**3220**
	Pe coloane	**3314**	**3282**
Accelerată	Pe linii	**2695**	2683
	Pe coloane	2722	2675
Iliffe	Pe linii	2731	**2525**
	Pe coloane	**2457**	**2419**

Dacă transpunem tabloul în felul următor: [99..101][-10..-9][-1..0][0..1][100..103][-5..4][-3..1]:

Tabela 2

Metoda	Reprezentare	Acces aliator (ms)	Acces liniar (ms)
Directă	Pe linii	**3255**	**3273**
	Pe coloane	**3284**	**3362**
Accelerată	Pe linii	2715	2712
	Pe coloane	**2756**	2730
Iliffe	Pe linii	**2404**	**2522**
	Pe coloane	2806	**2726**

Concluzii

Analizând tabelele 1 și 2: putem conclude că:

- Metoda directă – cea mai înceată, dar nu lasă amprentă de memorie;
- Metoda accelerată – viteză intermediară (nu mereu înceată decât vectorul Iliffe), dar lasă o amprentă mică de memorie;
- Metoda Iliffe – concomitent cea mai rapidă și mai înceată decât metoda accelerată, dar lasă o amprentă largă de memorie.

Viteza de acces unui element din tablou, utilizând vectorul lui Iliffe, depinde mult de numărul pointerilor alocate, cu atât mai puțini pointeri cu atât mai rapid este accesul.

Anexe

Se anexează codurile sursă elaborate în limbajul de programare C, în care sunt implementate metodele de acces cerute în sarcină.