

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХПІ»
Кафедра «Обчислювальна техніка та програмування»

Розрахункове завдання з дисципліни
«Алгоритми та структури даних»

Пояснювальна записка
КІТ 120-В 20.05.2021

Розробники

Виконав:

студент групи КІТ 120-В

Бабенко Антон Павлович

Перевірив:

Бульба Сергій Сергійович

Харків 2020

Розрахункове завдання з дисципліни
«Алгоритми та структури даних»

Пояснювальна записка
КІТ 120-В 20.05.2021

Листів 7

Харків 2020

1. Індивідуальне завдання:

Вектора Айліффа. Представити трьохвимірний масив за допомогою векторів Айліффа. Реалізувати функції ініціалізації, додавання, видалення елементів, звільнення пам'яті. Порівняти час доступу до елемента при поданні масива статичним вектором та векторами Айліффа.

2. Теоретичні відомості по темі завдання:

На логічному рівні двовимірний масив можна представити у вигляді таблиці, у якій довжина кожного рядка залишається незмінною по всьому масиву. Двовимірні масиви називають масивами масивів. Один із способів реалізації масивів з одним типом елементів на фізичному рівні (у внутрішній пам'яті) полягає у тому, що під масив виділяється безперервний блок пам'яті.

Перетворення логічної структури в фізичну називається процесом лінеаризації, у ході якого багатомірна логічна структура масива перетворюється в одновимірну фізичну структуру.

При звертанні до елемента двовимірного масива адреса елемента обчислюється так: $@a[i_1, i_2] = a_0 + (i_1 * N_1 + i_2) * S;$

для тривимірного: $@a[i_1, i_2, i_3] = a_0 + ((i_1 * N_1 + i_2) * N_2 + i_3) * S.$

Час доступу до елемента масива тим більше, чим більше вимірність масива.

Для зменшення часу доступу до елементів багатовимірних масивів їх представляють у вигляді векторів Айліффа. При цьому замість обчислення адреси виконується читання адрес векторів і елементів масиву.

3. Текст програми:

```
#include <ctime>
#include <iostream>
using namespace std;
```

```
const int N = 100;
const int M = 100;
const int K = 100;
```

```
int GetEl(int *arr, int i, int j, int k)
{
    return *(arr + ((K * M * i) + (K * j) + k));
}
```

```
void addIliffeEl(int i, int j, int k, int ***base, int data)
{
    *(*(*base + i) + j) + k) = data;
}
```

```
void removelliffeEl(int i, int j, int k, int ***base)
{
    *(*(*base + i) + j) + k) = 0;
}
```

```
int GetIliffeEl(int i, int j, int k, int ***base)
{
    return *(*(*base + i) + j) + k);
}
```

```
void freeMemory()
{
}
```

```
int main()
{
    int arr[N * N * N];
    for (int i = 0; i < N * N * N; i++)
    {
        arr[i] = i + 1;
    }
}
```

```
int i = 2;
int j = 2;
int k = 2;
```

```
unsigned int start_time = clock();
```

```
cout << GetEl(arr, i, j, k) << endl;
```

```
unsigned int end_time = clock();
```

```
unsigned int search_time = end_time - start_time;
```

```
cout << "Runtime: " << search_time / 10000.0 << " sec" << endl;
```

```
// ИНИЦИАЛИЗАЦИЯ ВЕКТОРОВ АЙЛИФФА
```

```
int ***base;
```

```
int **vec1[3];
```

```
base = vec1;
```

```
int count = 0;
```

```
for (int i = 0; i < N; i++)
```

```
{
```

```
    vec1[i] = new int *[M];
```

```
    for (int j = 0; j < M; j++)
```

```
    {
```

```
        vec1[i][j] = new int[K];
```

```
        for (int k = 0; k < K; k++)
```

```
        {
```

```
            vec1[i][j][k] = *(arr + count);
```

```
            count++;
```

```
        }
```

```
    }
```

```
}
```

```
////////////////////////////////////
```

```
start_time = clock();
```

```
cout << GetIliffeEl(i, j, k, base) << endl;
```

```
end_time = clock();
```

```

search_time = end_time - start_time;
cout << "Runtime: " << search_time / 10000.0 << " sec" << endl;
return 0;
}

```

4. Результати тестування:

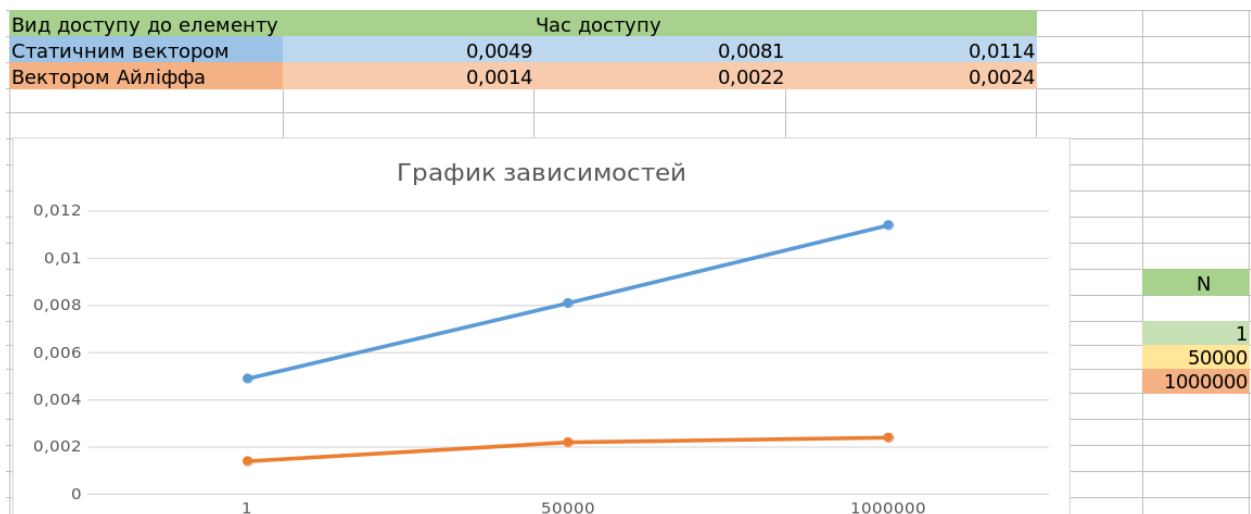
```

1
Runtime: 0.0049 sec
1
Runtime: 0.0014 sec
505051
Runtime: 0.0081 sec
505051
Runtime: 0.0022 sec
1000000
Runtime: 0.0114 sec
1000000
Runtime: 0.0024 sec

```

5. Пояснення результатів:

Як можемо бачити, програма створює статичний вектор. Потім представляє цей вектор як трьохвимірний масив за допомогою векторів Айліффа. Програму можна запустити декілька разів.



Я зробив заміри часу доступу до елементів за допомогою статичного вектора та вектором Айліффа. По графіку видно, що доступ до елементів за допомогою векторів Айліффа працює швидше, тому що скорочується час котрий витрачається на операцію множення. Але треба зазначити, що використання векторів Айліффа збільшує використання пам'яті, адже під кожний дескриптор виділяється окрема пам'ять.

6. Висновки:

В ході виконання цієї роботи я засвоїв роботу з багатовимірними масивами на логічному та фізичному рівні. Перетворення логічної структури в фізичну називається процесом лінеаризації. Для зменшення часу доступу до елементів багатовимірних масивів їх можна представити у вигляді векторів Айліффа.

Я розробив 2 методи доступу до елементів:

- Статичним вектором
- Векторами Айліффа