

НИУ ВШЭ ФКН

Архитектура Вычислительных Систем

Домашнее задание №4

Иванова Виолетта Александровна

Студентка БПИ-195(1)

ВАРИАНТ №11

## Условие задачи:

Определить индексы  $i, j$  ( $i \neq j$ ), для которых выражение  $A[i] - A[i+1] + A[i+2] - A[i+3] + \dots - A[j]$  имеет максимальное значение. Входные данные: массив чисел  $A$ , произвольной длины большей 10. Количество потоков не является параметром задачи.

## Ход решения:

Выписав формулу и рассмотрев некоторые варианты – я приняла решение использовать модель *итеративного параллелизма* в рамках решения этой задачи.

Потоки хранятся в векторе – индекс потока в векторе соответствует индексу  $i$  в массиве  $A$ , с которым этот поток будет работать.

Идея заключается в том, чтобы один поток брал одно значение  $A[i]$  и искал для него значения всех выражений с постепенным увеличением индекса  $j$ .

Результаты вычислений по итогам вычисления значения для *каждого* индекса сохраняются в векторе векторов чисел  $C$ . Так, например, выражение от  $A[i] \dots A[j]$  хранится в ячейке  $C[i][j]$ .

Благодаря такому алгоритму, потоки никак не влияют друг на друга, потому что каждый имеет уникальный индекс  $i$ .

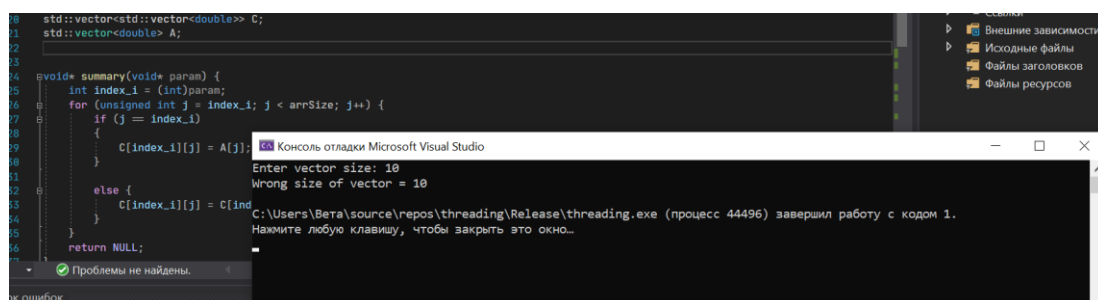
Чтобы дождаться завершения работы всех потоков пользуемся циклом `for`, но ожидаем потоки не с нулевого индекса, а с последнего, т. к. очевидно, что чем больше индекс  $i$ , тем меньше итераций пройдет цикл вычисления значения выражения.

После этого ищем наибольшее значение в матрице  $C$ , запоминая соответствующие индексы и выводим их в ответе.

Стоит сказать, что в программе стоит ограничение на максимальное количество потоков в 9999 элементов в массиве. Сделано это было, потому что потоков создаётся (количество элементов в массиве) – 1. Моя машина не выдерживает такие нагрузки.

Изначальный массив заполняется числами, которые соответствуют индексу элемента.

## Примеры работы программы на разных входных данных:



```
10 std::vector<std::vector<double>> C;
11 std::vector<double> A;
12
13 void* summary(void* param) {
14     int index_i = (int)param;
15     for (unsigned int j = index_i; j < arrSize; j++) {
16         if (j == index_i)
17             C[index_i][j] = A[j];
18         else {
19             C[index_i][j] = C[index_i][j-1] + A[j];
20         }
21     }
22     return NULL;
23 }
```

Консоль отладки Microsoft Visual Studio

Enter vector size: 10  
Wrong size of vector = 10

C:\Users\Beta\source\repos\threading\Release\threading.exe (процесс 44496) завершил работу с кодом 1.  
Нажмите любую клавишу, чтобы закрыть это окно...

Проблемы не найдены.

```
int main() {
    std::cout << "Enter vector size: ";
    std::cin >> arrSize;

    if (arrSize <= 10 || arrSize > 10000) {
        std::cout << "Wrong size of vector = " << arrSize << "\n";
        return 1;
    }

    A.resize(arrSize, 0);
    C.resize(arrSize, std::vector<double>(arrSize, DBL_MIN));

    for (unsigned int i = 0; i < arrSize; ++i) {
        A[i] = i;
    }

    for (unsigned int i = 0; i < arrSize; ++i) {
        {
            std::cout << "A[" << i << " = " << A[i] << " ";
            if (i % 10 == 9) std::cout << "\n";
        }
    }

    int threadNumber = arrSize / 10;
    std::vector<pthread_t> threads;

    for (int i = 0; i < threadNumber; ++i) {
        pthread_t t;
        pthread_create(&t, NULL, threadFunction, &A);
        threads.push_back(t);
    }

    pthread_join(threads[threadNumber - 1], NULL);

    std::cout << "Result indexes = " << A[threadNumber - 1] << " " << A[threadNumber] << "\n";

    C.resize(arrSize, std::vector<double>(arrSize, DBL_MIN));
}
```

Консоль отладки Microsoft Visual Studio

Enter vector size: 11

A[0] = 0  
A[1] = 1  
A[2] = 2  
A[3] = 3  
A[4] = 4  
A[5] = 5  
A[6] = 6  
A[7] = 7  
A[8] = 8  
A[9] = 9  
A[10] = 10  
Result indexes = 9 10

C:\Users\Beta\source\repos\threading\Release\threading.exe (процесс 12872) завершил работу с кодом 0.  
Нажмите любую клавишу, чтобы закрыть это окно...

0 Ошибки

Консоль отладки Microsoft Visual Studio

A[9974] = 9974  
A[9975] = 9975  
A[9976] = 9976  
A[9977] = 9977  
A[9978] = 9978  
A[9979] = 9979  
A[9980] = 9980  
A[9981] = 9981  
A[9982] = 9982  
A[9983] = 9983  
A[9984] = 9984  
A[9985] = 9985  
A[9986] = 9986  
A[9987] = 9987  
A[9988] = 9988  
A[9989] = 9989  
A[9990] = 9990  
A[9991] = 9991  
A[9992] = 9992  
A[9993] = 9993  
A[9994] = 9994  
A[9995] = 9995  
A[9996] = 9996  
A[9997] = 9997  
A[9998] = 9998  
Result indexes = 9997 9998

C:\Users\Beta\source\repos\threading\Release\threading.exe (процесс 18184) завершил работу с кодом 0.  
Нажмите любую клавишу, чтобы закрыть это окно...

Для демонстрации следующего примера генерация элементов была изменена:

```
if (arrSize <= 10 || arrSize > 10000) {
    std::cout << "Wrong size of vector = " << arrSize << "\n";
    return 1;
}

A.resize(arrSize, 0);
C.resize(arrSize, std::vector<double>(arrSize, DBL_MIN));

for (unsigned int i = 0; i < arrSize; ++i) {
    A[i] = rand();
    // A[i] = i;
}

for (unsigned int i = 0; i < arrSize; ++i) {
    std::cout << "A[" << i << " = " << A[i] << " ";
    if (i % 10 == 9) std::cout << "\n";
}

int threadNumber = arrSize / 10;
std::vector<pthread_t> threads;

for (int i = 0; i < threadNumber; ++i) {
    pthread_t t;
    pthread_create(&t, NULL, threadFunction, &A);
    threads.push_back(t);
}

pthread_join(threads[threadNumber - 1], NULL);

std::cout << "Result indexes = " << A[threadNumber - 1] << " " << A[threadNumber] << "\n";

C.resize(arrSize, std::vector<double>(arrSize, DBL_MIN));
}
```

Консоль отладки Microsoft Visual Studio

Enter vector size: 21

A[0] = 41  
A[1] = 18467  
A[2] = 6334  
A[3] = 26500  
A[4] = 19169  
A[5] = 15724  
A[6] = 11478  
A[7] = 29358  
A[8] = 26962  
A[9] = 24464  
A[10] = 5705  
A[11] = 28145  
A[12] = 23281  
A[13] = 16827  
A[14] = 9961  
A[15] = 491  
A[16] = 2995  
A[17] = 11942  
A[18] = 4827  
A[19] = 5436  
A[20] = 32391  
Result indexes = 11 20

C:\Users\Beta\source\repos\threading\Release\threading.exe (процесс 25544) завершил работу с кодом 0.  
Нажмите любую клавишу, чтобы закрыть это окно...

0 Ошибки

## Текст программы:

```
#include <iostream>
#include <vector>
#define HAVE_STRUCT_TIMESPEC
#include <pthread.h>

unsigned int arrSize;
std::vector<std::vector<double>> C;
std::vector<double> A;

void* summary(void* param) {
    int index_i = (int)param;
    for (unsigned int j = index_i; j < arrSize; j++) {
        if (j == index_i) {
            C[index_i][j] = A[j];
        }
        else {
            C[index_i][j] = C[index_i][j - 1] + A[j] * pow(-1, j);
        }
    }
    return NULL;
}

int main() {
    std::cout << "Enter vector size: ";
    std::cin >> arrSize;

    if (arrSize <= 10 || arrSize > 10000) {
        std::cout << "Wrong size of vector = " << arrSize << "\n";
        return 1;
    }

    A.resize(arrSize, 0);
    C.resize(arrSize, std::vector<double>(arrSize, DBL_MIN));

    for (unsigned int i = 0; i < arrSize; ++i) {
        // A[i] = rand();
        A[i] = i;
    }

    for (unsigned int i = 0; i < arrSize; i++) {
        std::cout << "A[" << i << "] = " << A[i] << "\n";
    }

    int threadNumber = arrSize - 1;
    std::vector<pthread_t> thread(threadNumber);

    for (int i = 0; i < threadNumber; i++) {
        pthread_create(&thread[i], NULL, summary, (void*)(i));
    }

    for (int i = threadNumber - 1; i > 0; i--) {
        pthread_join(thread[i], NULL);
    }

    double temp = DBL_MIN;
    int result_i = -1, result_j = -1;
    for (unsigned int i = 0; i < arrSize; i++) {
        for (unsigned int j = 0; j < arrSize; j++) {
            if (temp < C[i][j]) {
                temp = C[i][j];
                result_i = i;
                result_j = j;
            }
        }
    }

    std::cout << "Result indexes = " << result_i << " " << result_j << "\n";

    return 0;
}
```