

Никончик Даниил 12 группа 3 курс

Задание 10.12

Последовательная Программа

```
#include <vector>
#include <algorithm>
#include <iostream>
#include <random>
using namespace std; int main() {
    int rows = 100, cols = 100;

    mt19937 rng{ random_device{}() };
    uniform_int_distribution<int> dist{ 1, 100000 };

    std::vector<std::vector<int>> vec(rows, std::vector<int>(cols));
    for (auto i = 0; i < vec.size(); ++i)
        for (auto j = 0; j < vec[0].size(); ++j) vec[i][j] = dist(rng);

    std::vector<int> maximums(vec.size());
    for (auto i = 0; i < maximums.size(); ++i)
        maximums[i] = *std::max_element(vec[i].begin(), vec[i].end());
    int total_max = *std::max_element(maximums.begin(), maximums.end());
    std::cout << total_max << "\n";
    return 0;
}
```

//Параллельная

```
const int n_threads = 4;

mt19937 rng{ random_device{}() };
uniform_int_distribution<int> dist{ 1, 100000 };

vector<vector<int>> vec(rows, vector<int>(cols));
for (auto i = 0; i < cols; ++i)
    for (auto j = 0; j < rows; ++j) vec[i][j] = dist(rng);

////////////////////////////////////
auto start = std::chrono::high_resolution_clock::now();
vector<future<int>> futures;
vector<thread> threads;
vector<int> maximums(n_threads);

for (auto i = 0; i < n_threads; ++i) {
    auto task = packaged_task<int>(vector<vector<int>>&, int, int)>(subMatrixMax);
    futures.emplace_back(task.get_future());
    threads.emplace_back(move(task), ref(vec), i, n_threads);
}

for (auto i = 0; i < n_threads; ++i) {
    maximums[i] = futures[i].get(); threads[i].join();
}

auto end = std::chrono::high_resolution_clock::now();
auto res = std::chrono::duration_cast<std::chrono::microseconds>(end - start).count();
cout << "maximum = " << vector_max(maximums) << " time = " << res << "ms.\n";

}
```

Результаты 4 потока

Размерность	Линейная	Параллельная	Ускорение
1000*1000	45	31	1.45126129
10000*10000	273	53	5.1509434
1000000*1000000	2494	217	11.4930876