Дондик Ярослав, БПИ191

В данной папке представлено выполнение третьего домашнего задания по работе с параллельным программированием в C++.
Отчет лежит по этой ссылке - HW3/readme.pdf
Исходники лежат в HW3/src/, а скриншоты в HW3/img/

Вариант 14. Условие

Определить множество индексов i, для которых A[i] и B[i] не имеют общих делителей (единицу в роли делителя не рассматривать).
Входные данные: массивы целых положительных чисел A и B, произвольной длины ≥1000.
Количество потоков является входным параметром

Идея программы

- **Описание**
  Для выполнения задания был выбран следующий метод построения многопоточных приложений: итеративный параллелизм (все потоки работают над одной задачей-циклом)

Примерный алгоритм работы:

1. С консоли вводится размер массива и требуемое количество потоков
2. Создаются и заполняются два массива указанного размера
3. Запуск потоков, выполняющих свою часть цикла. Сохранение искомых индексов, вывод лог-данных в консоль
4. Вывод ответа и затраченное время на выполнение программы

Код программы

Исходный код
Исполняемый файл

Рассмотрим некоторые части кода, работу потоков и их основную функцию:

- Основная функция программы (main)

```cpp
// Инициализация массивов с числами
A = vector<int>(arrSize);
B = vector<int>(arrSize);

// Заполнение массивов числами до 100 000
for (int i = 0; i < arrSize; ++i) {
    A[i] = rand() % 100'000;
    B[i] = rand() % 100'000;
}

// Сохраняем время начала работы
auto begin = std::chrono::steady_clock::now();

// Создание потоков
for (int i = 0; i < threadNumber; i++) {
    thr.push_back(new thread{ gcd_cycle, i, threadNumber, ref(setIndexes) });
}

// Присоединение потоков
for (int i = 0; i < threadNumber; i++) {
    thr[i]->join();
    {
        // Блокировка вывода с других потоков
        unique_lock<mutex> locker(_lock);
        cout << "Thread " << i << " was closed\n";
    }
}
```

- Функция, которую выполняет каждый созданный поток

```cpp
/// <summary>
///  Функция, которую выполняют различные потоки. Каждый поток выполняет свою часть цикла (поставленную задачу)
/// </summary>
/// <param name="current_thread">Номер текущего потока</param>
/// <param name="count_threads">Общее количество потоков</param>
/// <param name="setIndex">Множество искомых индексов</param>
void gcd_cycle(int current_thread, int count_threads, set<int>& setIndex) {
    for (int i = current_thread; i < A.size(); i += count_threads) {

        // Подсчет наибольшего общего делителя
        unsigned gcd_result = GCD(A[i], B[i]);

        // Если нет общих делителей, кроме 1:
        if (gcd_result == 1) {
            {
                // Блокировка вывода с других потоков
                unique_lock<mutex> locker(_lock);
                cout << "Thread " << current_thread << " found pair of gcd == 1:" <<
                    " A[" << i << "] = " << A[i] << " and B[" << i << "] = " << B[i] << "\n";
            }
            // Добавляем индекс в наше множество
            setIndex.insert(i);
        }
    }
}
```

- Функция поиска наибольшего общего делителя двух чисел (основана на алгоритме Евклида)

```
/// <summary>
/// Вычисление наибольшего общего делителя
/// </summary>
/// <param name="u">Первое число</param>
/// <param name="v">Второе число</param>
/// <returns>Наибольший общий делитель переданных чисел</returns>
unsigned GCD(unsigned u, unsigned v) {
    while (v != 0) {
        unsigned r = u % v;
        u = v;
        v = r;
    }
    return u;
}
```

Тестирование программы

- **Тест 1**

  Проверим программу на некорректном вводе:

  Консоль отладки Microsoft Visual Studio

  ```
  Please, enter size of an array of integers (>= 1000): qwerty
  Incorrect input. Please, enter your number again: -10
  Incorrect input. Please, enter your number again: -1
  Incorrect input. Please, enter your number again: 0
  Incorrect input. Please, enter your number again: 999
  Incorrect input. Please, enter your number again: 1000
  Please, enter number of threads: qwerty
  Incorrect input. Please, enter your number again: -1
  Incorrect input. Please, enter your number again: 0
  Incorrect input. Please, enter your number again: 5
  Thread 0 found pair of gcd == 1: A[0] = 28769 and B[0] = 16547
  Thread 0 found pair of gcd == 1: A[15] = 325 and B[15] = 30709
  Thread 0 found pair of gcd == 1: A[20] = 1569 and B[20] = 12842
  Thread 0 found pair of gcd == 1: A[25] = 24939 and B[25] = 5579
  Thread 0 found pair of gcd == 1: A[30] = 619 and B[30] = 10263
  Thread 1 found pair of gcd == 1: A[1] = 28269 and B[1] = 22799
  Thread 1 found pair of gcd == 1: A[6] = 30737 and B[6] = 19198
  ```

  После успешного ввода происходит вывод логов от каждого потока:

```
Thread 2 found pair of gcd == 1: A[322] = 4740 and B[322] = 16129
Thread 2 found pair of gcd == 1: A[327] = 31161 and B[327] = 4088
Thread 2 found pair of gcd == 1: A[357] = 26636 and B[357] = 31723
Thread 2 found pair of gcd == 1: A[367] = 24181 and B[367] = 30971
Thread 2 found pair of gcd == 1: A[377] = 7911 and B[377] = 31129
Thread 3 found pair of gcd == 1: A[8] = 16285 and B[8] = 9006
Thread 4 found pair of gcd == 1: A[4] = 3539 and B[4] = 1523
Thread 4 found pair of gcd == 1: A[9] = 9996 and B[9] = 7027
Thread 4 found pair of gcd == 1: A[19] = 14295 and B[19] = 17231
Thread 4 found pair of gcd == 1: A[24] = 5249 and B[24] = 27251
Thread 4 found pair of gcd == 1: A[39] = 1318 and B[39] = 16783
Thread 4 found pair of gcd == 1: A[49] = 7255 and B[49] = 27021
Thread 4 found pair of gcd == 1: A[64] = 7095 and B[64] = 866
Thread 4 found pair of gcd == 1: A[74] = 7029 and B[74] = 30073
Thread 3 found pair of gcd == 1: A[13] = 1647 and B[13] = 3784
Thread 3 found pair of gcd == 1: A[23] = 31309 and B[23] = 20302
Thread 1 found pair of gcd == 1: A[31] = 20765 and B[31] = 4253
Thread 1 found pair of gcd == 1: A[36] = 5713 and B[36] = 17913
Thread 1 found pair of gcd == 1: A[41] = 11749 and B[41] = 2833
Thread 1 found pair of gcd == 1: A[46] = 23572 and B[46] = 19269
Thread 3 found pair of gcd == 1: A[33] = 21228 and B[33] = 1657
Thread 3 found pair of gcd == 1: A[43] = 743 and B[43] = 28880
Thread 4 found pair of gcd == 1: A[89] = 584 and B[89] = 1415
Thread 4 found pair of gcd == 1: A[94] = 589 and B[94] = 15505
Thread 1 found pair of gcd == 1: A[51] = 5687 and B[51] = 26216
Thread 1 found pair of gcd == 1: A[71] = 12320 and B[71] = 3579
Thread 3 found pair of gcd == 1: A[53] = 22633 and B[53] = 5407
```

Далее все потоки закрываются и выводится ответ:

```
Thread 0 was closed
Thread 1 was closed
Thread 2 was closed
Thread 3 was closed
Thread 4 was closed

Set of indexes:
0 1 4 6 8 9 11 13 15 16 17 19 20 23 24 25 26 27 30 31 33 35 36 37 39 41 43 45 46 49 51 53 57 58 62 64 65 67 68 71 72 73 74 76 77 78
80 81 82 86 87 88 89 90 92 93 94 98 99 100 101 105 106 107 108 109 111 113 114 115 116 117 119 120 122 123 125 131 134 136 138 141 1
42 143 144 145 146 147 148 149 150 151 152 153 156 157 159 160 161 163 164 165 175 176 177 178 179 180 181 184 186 187 188 190 191 1
92 194 195 198 199 202 203 205 206 207 208 209 210 211 213 216 217 219 220 221 223 224 225 226 228 230 231 233 234 236 237 238 240 2
42 243 244 246 247 248 249 250 252 254 255 257 259 260 262 263 264 265 268 269 270 271 272 277 279 280 281 284 285 286 287 289 293 2
94 296 297 299 301 302 306 308 309 310 314 315 316 317 318 320 321 322 323 325 327 328 329 330 333 334 335 336 339 340 341 345 346 3
48 349 351 354 357 358 365 366 367 368 369 370 373 374 377 378 379 380 381 384 385 386 387 388 390 391 394 395 396 398 399 400 401 4
04 406 407 408 409 411 414 415 417 418 419 420 422 423 426 428 429 432 433 434 435 436 437 438 440 442 443 445 449 450 451 452 453 4
55 456 457 460 461 463 466 468 469 471 474 475 479 480 481 482 483 487 489 491 499 500 501 502 506 507 508 509 510 512 514 515 516 5
20 523 524 525 526 529 530 532 535 536 537 539 540 541 544 545 546 547 548 553 556 563 564 566 570 571 572 574 575 576 579 580 581 5
26 627 630 631 632 636 637 638 639 640 641 642 643 644 645 649 650 651 653 655 658 659 660 661 662 665 668 670 672 673 674 678 680 6
83 584 585 588 589 590 591 592 593 594 595 596 598 601 602 604 605 607 608 609 610 611 612 614 615 617 618 619 620 622 623 624 625 6
81 685 686 687 689 692 693 694 696 697 698 700 701 702 703 706 707 708 710 711 712 713 714 716 717 718 719 722 724 725 729 731 732 7
33 734 735 737 738 739 740 741 742 743 744 745 747 748 749 750 751 753 754 757 758 759 761 762 763 765 766 768 770 774 776 778 780 7
81 784 785 787 788 791 792 793 794 795 796 797 799 800 801 802 803 805 806 807 808 809 810 811 813 814 815 816 817 820 826 827 829 8
31 832 834 837 839 840 843 844 845 847 848 849 850 854 858 861 862 863 864 866 867 868 870 873 874 875 876 877 878 880 881 884 885 8
86 890 891 892 894 898 900 901 903 904 908 909 910 912 913 914 915 916 917 918 920 922 924 926 927 930 933 934 936 937 939 940 941 9
42 946 947 948 949 953 954 955 956 957 958 959 960 963 964 965 967 968 969 972 973 974 975 976 977 978 985 986 987 990 992 993 994 9
96 997 998 999

Elapsed time: 1831 ms
```

Программа отработала успешно.

- **Тест 2**

  Проверим программу с большим количеством потоков (10):

```
Please, enter size of an array of integers (>= 1000): 1000
Please, enter number of threads: 10
```

Вывод логов от каждого потока. Некоторые потоки заканчивают свою работу раньше других:

```
Thread 1 found pair of gcd == 1: A[971] = 3165 and B[971] = 14377
Thread 1 was closed
Thread 2 was closed
Thread 3 found pair of gcd == 1: A[673] = 7023 and B[673] = 3143
Thread 3 found pair of gcd == 1: A[703] = 15055 and B[703] = 8749
Thread 3 found pair of gcd == 1: A[713] = 6625 and B[713] = 5266
Thread 3 found pair of gcd == 1: A[733] = 15415 and B[733] = 22021
Thread 3 found pair of gcd == 1: A[753] = 27563 and B[753] = 4068
Thread 3 found pair of gcd == 1: A[803] = 32201 and B[803] = 26940
Thread 3 found pair of gcd == 1: A[813] = 9857 and B[813] = 15271
Thread 3 found pair of gcd == 1: A[823] = 9115 and B[823] = 29982
Thread 3 found pair of gcd == 1: A[863] = 29951 and B[863] = 46
Thread 3 found pair of gcd == 1: A[873] = 19843 and B[873] = 29389
Thread 3 found pair of gcd == 1: A[883] = 28465 and B[883] = 25723
Thread 3 found pair of gcd == 1: A[893] = 16246 and B[893] = 29067
Thread 3 found pair of gcd == 1: A[903] = 31381 and B[903] = 29368
Thread 3 found pair of gcd == 1: A[913] = 11863 and B[913] = 7490
Thread 3 found pair of gcd == 1: A[923] = 15589 and B[923] = 2614
Thread 3 found pair of gcd == 1: A[943] = 17438 and B[943] = 24955
Thread 3 found pair of gcd == 1: A[953] = 17560 and B[953] = 10367
Thread 3 found pair of gcd == 1: A[963] = 12035 and B[963] = 3263
Thread 4 found pair of gcd == 1: A[184] = 9653 and B[184] = 11049
Thread 4 found pair of gcd == 1: A[194] = 32471 and B[194] = 21941
Thread 4 found pair of gcd == 1: A[234] = 20115 and B[234] = 5731
Thread 4 found pair of gcd == 1: A[244] = 22580 and B[244] = 2127
Thread 4 found pair of gcd == 1: A[264] = 32336 and B[264] = 6693
Thread 4 found pair of gcd == 1: A[284] = 14073 and B[284] = 8824
Thread 3 was closed
```

Завершаются оставшиеся потоки и выводится ответ:

```
Консоль отладки Microsoft Visual Studio                                    —    □    ×
Thread 7 found pair of gcd == 1: A[907] = 14518 and B[907] = 18211
Thread 7 found pair of gcd == 1: A[947] = 9103 and B[947] = 5597
Thread 7 found pair of gcd == 1: A[987] = 18266 and B[987] = 24529
Thread 6 was closed
Thread 7 was closed
Thread 8 was closed
Thread 9 was closed

Set of indexes:
0 2 4 6 8 10 11 12 13 15 16 17 18 20 23 25 26 29 31 37 38 39 42 43 44 45 46 48 49 50 51 53 54 57 58 59 60 61 62 65 66 68 71 72 75 76 77 81 8
5 86 87 91 92 93 95 99 102 109 111 113 114 115 116 119 121 123 128 129 131 132 134 137 138 140 141 142 144 145 146 147 148 150 151 154 155 1
56 157 158 160 161 162 163 164 165 166 168 169 170 171 173 174 177 178 179 180 181 182 183 184 186 188 189 190 191 192 194 197 199 202 205 2
06 207 209 210 213 215 216 217 218 220 221 222 223 226 227 229 230 231 232 233 234 235 236 239 240 242 243 244 245 246 247 248 249 250 251 2
52 255 257 258 259 260 261 262 263 264 265 266 267 268 272 273 276 277 280 284 287 288 289 291 294 295 296 297 298 299 300 301 302 303 304 3
05 306 307 308 310 312 313 314 317 318 322 324 325 326 327 328 331 333 334 335 338 339 341 343 344 347 350 351 357 358 360 361 363 364 366 3
67 368 370 371 372 373 375 376 377 378 380 384 386 387 389 394 396 397 399 400 401 402 404 405 406 407 408 411 412 416 417 419 420 421 424 4
25 426 427 428 429 432 433 435 436 439 441 442 445 447 449 450 451 452 453 454 455 456 458 460 462 465 466 469 470 473 474 476 477 479 482 4
83 484 486 490 494 495 496 500 501 502 505 506 508 510 511 512 513 514 516 517 518 520 521 523 524 525 527 528 530 531 532 533 534 536 537 5
39 543 545 546 547 548 549 550 551 552 553 554 560 561 562 564 565 566 567 568 569 570 572 573 578 579 582 583 585 586 588 590 591 593 5
95 598 599 600 601 602 603 606 607 608 609 610 612 613 615 616 617 618 619 620 621 624 626 629 634 636 637 638 640 641 642 643 644 645 648 6
49 651 653 657 658 664 665 666 668 669 671 672 673 674 676 677 678 680 684 685 686 687 688 689 690 691 692 694 696 699 700 701 702 703 7
04 706 710 713 715 716 717 720 721 722 724 725 726 727 730 732 733 737 738 740 741 742 744 746 747 748 751 752 753 754 756 759 761 762 764 7
65 766 767 768 770 771 772 780 781 782 784 785 788 789 790 792 795 798 800 802 803 804 805 807 808 811 812 813 814 816 818 819 820 821 822 8
23 824 826 827 829 831 832 834 835 836 838 844 845 846 848 852 855 856 859 860 863 865 866 867 868 870 871 873 876 878 879 882 883 885 886 8
87 889 892 893 894 896 897 898 899 901 903 905 906 907 909 911 912 913 915 916 922 923 924 925 929 930 931 934 936 938 939 940 943 944 947 9
48 949 950 952 953 955 956 958 959 960 962 963 965 971 972 974 975 976 978 979 980 982 984 985 986 987 988 989 990 992 995 996 998 999

Elapsed time: 1488 ms
```

Программа отработала успешно.

- **Тест 3**

  Проверим программу с еще большим количеством потоков (1000):

Завершение работы потоков и вывод ответа:

Программа отработала успешно.

- **Тест 4**

  Проверим программу введем размер массива и количество потоков, равными 100000 и 10000 соответственно:

```
Please, enter size of an array of integers (>= 1000): 100000
Please, enter number of threads: 10000
Thread 0 found pair of gcd == 1: A[0] = 10497 and B[0] = 18515
Thread 0 found pair of gcd == 1: A[10000] = 28895 and B[10000] = 13791
Thread 0 found pair of gcd == 1: A[20000] = 20017 and B[20000] = 17745
Thread 0 found pair of gcd == 1: A[40000] = 11727 and B[40000] = 7172
Thread 0 found pair of gcd == 1: A[60000] = 14619 and B[60000] = 30892
Thread 0 found pair of gcd == 1: A[80000] = 8533 and B[80000] = 18314
Thread 6 found pair of gcd == 1: A[6] = 1207 and B[6] = 2483
Thread 6 found pair of gcd == 1: A[10006] = 18255 and B[10006] = 682
Thread 6 found pair of gcd == 1: A[30006] = 32023 and B[30006] = 15759
Thread 6 found pair of gcd == 1: A[50006] = 22537 and B[50006] = 501
Thread 6 found pair of gcd == 1: A[60006] = 26646 and B[60006] = 2509
Thread 6 found pair of gcd == 1: A[80006] = 27039 and B[80006] = 12269
Thread 4 found pair of gcd == 1: A[20004] = 27253 and B[20004] = 22626
Thread 4 found pair of gcd == 1: A[30004] = 215 and B[30004] = 6513
Thread 4 found pair of gcd == 1: A[40004] = 20039 and B[40004] = 32663
Thread 4 found pair of gcd == 1: A[50004] = 29677 and B[50004] = 18268
Thread 4 found pair of gcd == 1: A[60004] = 5872 and B[60004] = 8521
Thread 16 found pair of gcd == 1: A[40016] = 14805 and B[40016] = 28862
Thread 15 found pair of gcd == 1: A[15] = 17711 and B[15] = 9165
Thread 15 found pair of gcd == 1: A[30015] = 15467 and B[30015] = 335
Thread 9 found pair of gcd == 1: A[9] = 10095 and B[9] = 1421
Thread 9 found pair of gcd == 1: A[10009] = 29607 and B[10009] = 27490
Thread 9 found pair of gcd == 1: A[50009] = 25789 and B[50009] = 11402
Thread 23 found pair of gcd == 1: A[20023] = 12377 and B[20023] = 19335
Thread 612 found pair of gcd == 1: A[612] = 10211 and B[612] = 9265
Thread 612 found pair of gcd == 1: A[20612] = 18514 and B[20612] = 10129
```

завершение всех потоков:

```
Thread 9979 was closed
Thread 9980 was closed
Thread 9981 was closed
Thread 9982 was closed
Thread 9983 was closed
Thread 9984 was closed
Thread 9985 was closed
Thread 9986 was closed
Thread 9987 was closed
Thread 9988 was closed
Thread 9989 was closed
Thread 9990 was closed
Thread 9991 was closed
Thread 9992 was closed
Thread 9993 was closed
Thread 9994 was closed
Thread 9995 was closed
Thread 9996 was closed
Thread 9997 was closed
Thread 9998 was closed
Thread 9999 was closed

Set of indexes:
2 3 4 7 9 10 13 15 16 18 19 20 22
 73 74 76 77 78 81 82 84 86 89 90
```

Вывод ответа. Длительное время работы программы объясняется медленной работой вывода большого количества строчек в консоль:

```
6 99237 99238 99239 99240 99242 99245 99246 99247 99249 99250 99254 99255 99256 99257 99258 99262 99263 99264 99268 9927
2 99275 99277 99278 99279 99280 99281 99282 99284 99286 99289 99291 99292 99294 99296 99297 99299 99301 99302 99303 9930
4 99305 99308 99309 99310 99311 99313 99314 99315 99316 99317 99318 99319 99320 99321 99323 99324 99325 99327 99328 9932
9 99330 99334 99335 99336 99339 99340 99341 99343 99344 99345 99346 99347 99348 99352 99353 99354 99355 99358 99360 9936
1 99362 99364 99366 99369 99372 99373 99374 99377 99378 99381 99382 99385 99386 99387 99389 99391 99392 99394 99395 9939
7 99400 99402 99403 99404 99405 99406 99407 99410 99413 99414 99418 99419 99420 99421 99422 99424 99427 99428 99431 9943
4 99436 99438 99442 99444 99447 99453 99454 99455 99459 99460 99461 99464 99468 99475 99476 99477 99481 9948
2 99484 99485 99490 99491 99493 99495 99498 99500 99501 99502 99503 99504 99505 99507 99508 99509 99510 99512 99513 9951
4 99515 99519 99520 99521 99522 99523 99525 99526 99528 99532 99534 99536 99539 99540 99541 99542 99543 99544 99547 9954
8 99551 99554 99555 99556 99557 99558 99559 99561 99562 99563 99564 99565 99566 99568 99571 99572 99573 99575 99576 9957
8 99582 99586 99588 99590 99591 99592 99594 99595 99596 99597 99599 99601 99603 99605 99606 99607 99609 99610 9961
3 99615 99616 99618 99621 99622 99623 99624 99625 99629 99630 99632 99633 99635 99636 99637 99638 99639 99640 99642 9964
3 99644 99646 99647 99651 99654 99658 99659 99661 99662 99664 99665 99668 99670 99671 99672 99673 99674 99675 99676 9967
7 99678 99682 99686 99687 99688 99690 99691 99695 99696 99699 99700 99701 99703 99706 99707 99710 99711 99712 99713 9971
4 99715 99716 99719 99720 99722 99724 99725 99727 99730 99732 99734 99736 99737 99739 99740 99741 99742 99745 99747 9974
8 99749 99750 99751 99754 99756 99757 99758 99759 99760 99764 99766 99767 99769 99770 99771 99774 99776 99777 99778 9977
9 99780 99781 99782 99784 99788 99789 99790 99791 99792 99793 99795 99796 99798 99799 99800 99801 99804 99806 99807 9981
0 99811 99812 99814 99816 99817 99819 99820 99822 99823 99828 99829 99831 99833 99834 99835 99836 99838 99839 99840 9984
1 99842 99843 99845 99847 99848 99850 99851 99852 99857 99859 99861 99862 99863 99864 99865 99868 99870 99872 99873 9987
4 99875 99876 99877 99878 99879 99881 99883 99884 99885 99886 99887 99888 99889 99890 99891 99892 99893 99896 9989
7 99898 99902 99903 99904 99905 99906 99908 99909 99910 99911 99915 99916 99919 99920 99921 99922 99923 99924 99925 9992
6 99928 99931 99932 99933 99938 99940 99941 99942 99943 99945 99946 99947 99948 99956 99960 99961 99962 99963 99966 9996
8 99969 99970 99973 99974 99975 99976 99978 99981 99984 99987 99988 99990 99991 99993 99994 99996 99999

Elapsed time: 99557 ms
```

Программа отработала успешно.

Источники информации

1. Парадигмы параллельного программирования [Электронный ресурс] // Блог программиста : [сайт]. [2020]. URL: https://pro-prof.com/forums/topic/parallel-programming-paradigms, режим доступа: свободный, дата обращения: 17.11.2020

2. Работа с потоками, мьютексами в C++ [Электронный ресурс] // Habr: [сайт]. [2020]. URL: https://habr.com/ru/post/182610/, режим доступа: свободный, дата обращения: 17.11.2020