Министерство образования Республики Беларусь

Учреждения образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине МОИС за 1 семестр

«Группы»

Выполнил

Студент группы ИИ-23

Романюк А. П.

Проверил

Козинский А.А.

Брест 2023

***Задание 1.***

1. Представить заданную подстановку произведением независимых циклов. Варианты заданий указаны в таблице 1. Подстановка из 6 элементов задана в таблице 1 второй строкой подстановки. Например: подстановка записана в таблице как .
2. Определить степень заданной подстановки.
3. Определить четность подстановки:  
   - через число транспозиций в разложении подстановки в произведение транспозиций;  
   - через число инверсий в подстановке.
4. Задания реализовать программно.

#include <iostream>

#include <vector>

#include <fstream>

#include <vector>

using namespace std;

void find\_cycle(vector<int> subst) {

ofstream out("output.txt", ios::app);

bool\* check = new bool[subst.size()] {0};

int j = 1;

for (int i = 0; i < subst.size(); i++) {

if (!check[i]) {

out << "cucle " << j << ":(" << i + 1;

int pos = subst[i];

check[i] = 1;

while (pos != i) {

out << pos + 1;

check[pos] = 1;

pos = subst[pos];

check[pos] = 1;

}

out << ")" << endl;

j++;

}

}

}

void find\_degree(vector<int> subst) {

ofstream out("output.txt", ios::app);

out << "degree of substitution:" << subst.size() << endl;

}

void invers\_chet(vector<int> subst) {

ofstream out("output.txt", ios::app);

int inv = 0;

for (int i = 0; i < subst.size()-1; i++) {

for (int j = i + 1; j < subst.size(); j++) {

if (subst[i] > subst[j]) inv++;

}

}

out << "parity of substitution by inversions: ";

if (inv % 2 == 0) out << "true\n"; else out << "false\n";

}

void transpos\_chet(const vector<int> subst) {

ofstream out("output.txt", ios::app);

int n = subst.size();

vector<bool> visited(n, false);

int transpositions = 0;

for (int i = 0; i < n; i++) {

if (!visited[i]) {

int j = i;

int cycle\_size = 0;

while (!visited[j]) {

visited[j] = true;

j = subst[j];

cycle\_size++;

}

transpositions += cycle\_size - 1;

}

}

out << "parity of substitution by transpositions: ";

if (transpositions % 2 == 0) out << "true\n"; else out << "false\n";

}

int main() {

ofstream out("output.txt");

ifstream in("input.txt");

vector<int> subst;

while (!in.eof()) {

int buf;

in >> buf;

subst.push\_back(buf-1);

}

find\_cycle(subst);

find\_degree(subst);

invers\_chet(subst);

transpos\_chet(subst);

}

cucle 1:(1426)

cucle 2:(3)

cucle 3:(5)

degree of substitution:6

parity of substitution by inversions: false

parity of substitution by transpositions: false

***Задание 2.***

Для заданной в таблице 1 группы:

1. Построить таблицу Кэли. Определить нейтральный элемент. Для каждого элемента группы указать обратный элемент.
2. Определить является ли группа циклической, указать порядок каждого элемента в группе.
3. Указать все подгруппы заданной группы и их порядок. Определить порождающие элементы для подгрупп. Для одной из подгрупп построить таблицу Кэли.
4. Разложить группу на левые смежные классы по каждой из подгрупп. Построить фактор множества для группы по каждой из подгрупп.
5. Построить таблицу Кэли для фактор-группы по одной из подгрупп.
6. #include <iostream>
7. #include <vector>
8. #include <iomanip>
9. #include <algorithm>
10. #include <set>
11. using namespace std;
12. vector<int> group = { 1, 2, 4, 8 };
13. vector<vector<int>> subgroups = { {1}, {1, 2}, {1, 4}, {1, 8}, {1, 2, 4, 8} };
14. set<int> left\_coset(int h, const vector<int>& subgroup) {
15. set<int> coset;
16. for (int g : subgroup) {
17. coset.insert((h \* g) % 15);
18. }
19. return coset;
20. }
21. void print\_set(const set<int>& s) {
22. cout << "{ ";
23. for (int x : s) {
24. cout << x << " ";
25. }
26. cout << "}";
27. }
28. int main() {
29. vector<int> A = { 1, 2, 4, 8 };
30. vector<vector<int>> AK(A.size(), vector<int>(A.size()));
31. cout << setw(2) << " \*|";
32. for (int j = 0; j < A.size(); j++) {
33. cout << setw(2) << A[j] << " ";
34. }
35. cout << endl;
36. cout << " \_+\_\_\_\_\_\_\_\_\_\_\_\n";
37. for (int i = 0; i < A.size(); i++) {
38. cout << setw(2) << A[i] << "|";
39. for (int j = 0; j < A.size(); j++) {
40. AK[i][j] = A[i] \* A[j] % 15;
41. cout << setw(2) << AK[i][j] << " ";
42. }
43. cout << endl;
44. }
45. cout << "neytrall element: 1\n";
46. cout << "inverse elements:\n";
47. for (int i = 0; i < A.size(); i++) {
48. for (int j = 0; j < A.size(); j++) {
49. if (AK[i][j] == 1) {
50. cout << A[i] << ":" << A[j] << endl;
51. }
52. }
53. }
54. cout << "Groups:\n";
55. vector<vector<int>> groups(A.size(), vector<int>(A.size()));
56. bool check = 0;
57. for (int i = 0; i < A.size(); i++) {
58. int count = 0;
59. cout << A[i] << ":{ ";
60. int k = A[i];
61. do {
62. k = A[i]\*k%15;
63. groups[i][count] = k;
64. cout << k << " ";
65. count++;
66. } while (k != A[i]);
67. cout << "} order of an element: " << count << endl;
68. if (count == A.size()) check = 1;
69. }
70. cout << "Group is cyclic: ";
71. if (check) cout << "TRUE\n"; else cout << "FALSE\n";
72. cout << setw(2) << " \*|";
73. for (int j = 0; j < 2; j++) {
74. cout << setw(2) << groups[2][j] << " ";
75. }
76. cout << endl;
77. cout << " \_+\_\_\_\_\_\_\_\_\_\_\_\n";
78. for (int i = 0; i < 2; i++) {
79. cout << setw(2) << groups[2][i] << "|";
80. for (int j = 0; j < 2; j++) {
81. cout << setw(2) << groups[2][i]\* groups[2][j]%15 << " ";
82. }
83. cout << endl;
84. }
85. for (const vector<int>& H : subgroups) {
86. cout << "Subgroup: ";
87. print\_set(set<int>(H.begin(), H.end()));
88. cout << "\nLeft cosets:\n";
89. set<set<int>> cosets;
90. for (int h : group) {
91. set<int> coset = left\_coset(h, H);
92. if (!cosets.count(coset)) {
93. cosets.insert(coset);
94. print\_set(coset);
95. cout << "\n";
96. }
97. }
98. cout << "Factor set: \n";
99. for (const auto& coset : cosets) {
100. print\_set(coset);
101. cout << "\n";
102. }
103. cout << "\n";
104. }
106. return 0;
107. }

\*| 1 2 4 8

\_+\_\_\_\_\_\_\_\_\_\_\_

1| 1 2 4 8

2| 2 4 8 1

4| 4 8 1 2

8| 8 1 2 4

neytrall element: 1

inverse elements:

1:1

2:8

4:4

8:2

Groups:

1:{ 1 } order of an element: 1

2:{ 4 8 1 2 } order of an element: 4

4:{ 1 4 } order of an element: 2

8:{ 4 2 1 8 } order of an element: 4

Group is cyclic: TRUE

\*| 1 4

\_+\_\_\_\_\_\_\_\_\_\_\_

1| 1 4

4| 4 1

Subgroup: { 1 }

Left cosets:

{ 1 }

{ 2 }

{ 4 }

{ 8 }

Factor set:

{ 1 }

{ 2 }

{ 4 }

{ 8 }

Subgroup: { 1 2 }

Left cosets:

{ 1 2 }

{ 2 4 }

{ 4 8 }

{ 1 8 }

Factor set:

{ 1 2 }

{ 1 8 }

{ 2 4 }

{ 4 8 }

Subgroup: { 1 4 }

Left cosets:

{ 1 4 }

{ 2 8 }

Factor set:

{ 1 4 }

{ 2 8 }

Subgroup: { 1 8 }

Left cosets:

{ 1 8 }

{ 1 2 }

{ 2 4 }

{ 4 8 }

Factor set:

{ 1 2 }

{ 1 8 }

{ 2 4 }

{ 4 8 }

Subgroup: { 1 2 4 8 }

Left cosets:

{ 1 2 4 8 }

Factor set:

{ 1 2 4 8 }

Пусть H будет подгруппой, в данном случае H = {1}, и пусть x принадлежит группе G, где G = {1, 2, 4, 8}. Тогда левый смежный класс будет выглядеть следующим образом:

1H = {1 \* 1} = {1} 2H = {2 \* 1} = {2} 4H = {4 \* 1} = {4} 8H = {8 \* 1} = {8}

| 1H 2H 4H 8H

1H | 1H 2H 4H 8H

2H | 2H 4H 8H 1H

4H | 4H 8H 1H 2H

8H | 8H 1H 2H 4H

***Задание 3.***

Построить гомоморфизм групп . Дать гомоморфизму характеристику:

;

Мы можем определить гомоморфизм f: G → G’ следующим образом: f(a) = a mod 4 + 1.

- ; :

f(a) = 3^a.

Характеристики гомоморфизмов:

Сохранение операции

Сохранение нейтрального элемента

Сохранение обратного элемента