MAT3_LAB - Tymon Zadara

Kwadrat:

In[9]:=
```
DihedralGroup[4] // GroupOrder
```
Out[9]=
```
8
```

In[3]:=
```
D4 = DihedralGroup[4] // GroupElements
```
Out[3]=
```
{Cycles[{}], Cycles[{{2, 4}}], Cycles[{{1, 2}, {3, 4}}], Cycles[{{1, 2, 3, 4}}],
 Cycles[{{1, 3}}], Cycles[{{1, 3}, {2, 4}}], Cycles[{{1, 4, 3, 2}}], Cycles[{{1, 4}, {2, 3}}]}
```

In[5]:=
```
Map[PermutationList[#, 4] &, D4]
```
Out[5]=
```
{{1, 2, 3, 4}, {1, 4, 3, 2}, {2, 1, 4, 3},
 {2, 3, 4, 1}, {3, 2, 1, 4}, {3, 4, 1, 2}, {4, 1, 2, 3}, {4, 3, 2, 1}}
```

Pięciokąt:

In[11]:=
```
DihedralGroup[5] // GroupOrder
```
Out[11]=
```
10
```

In[12]:=
```
D5 = DihedralGroup[5] // GroupElements
```
Out[12]=
```
{Cycles[{}], Cycles[{{2, 5}, {3, 4}}], Cycles[{{1, 2}, {3, 5}}], Cycles[{{1, 2, 3, 4, 5}}],
 Cycles[{{1, 3}, {4, 5}}], Cycles[{{1, 3, 5, 2, 4}}], Cycles[{{1, 4}, {2, 3}}],
 Cycles[{{1, 4, 2, 5, 3}}], Cycles[{{1, 5, 4, 3, 2}}], Cycles[{{1, 5}, {2, 4}}]}
```

In[13]:=
```
Map[PermutationList[#, 5] &, D5]
```
Out[13]=
```
{{1, 2, 3, 4, 5}, {1, 5, 4, 3, 2}, {2, 1, 5, 4, 3}, {2, 3, 4, 5, 1}, {3, 2, 1, 5, 4},
 {3, 4, 5, 1, 2}, {4, 3, 2, 1, 5}, {4, 5, 1, 2, 3}, {5, 1, 2, 3, 4}, {5, 4, 3, 2, 1}}
```

Czworościan Foremny:

!!! AlternatingGroup - funkcja znajdująca grupę parzystych permutacji pewnego zbioru !!!

W zadaniu jest dane: Grupa jest izomorficzna z grupa(A4, ∘). Stąd wykorzystanie funkcji

In[15]:=
```
AlternatingGroup[4] // GroupOrder
```

Out[15]=
```
12
```

In[25]:=
```
D4A = AlternatingGroup[4] // GroupElements
```

Out[25]=
```
{Cycles[{}], Cycles[{{2, 3, 4}}], Cycles[{{2, 4, 3}}], Cycles[{{1, 2}, {3, 4}}],
 Cycles[{{1, 2, 3}}], Cycles[{{1, 2, 4}}], Cycles[{{1, 3, 2}}], Cycles[{{1, 3, 4}}],
 Cycles[{{1, 3}, {2, 4}}], Cycles[{{1, 4, 2}}], Cycles[{{1, 4, 3}}], Cycles[{{1, 4}, {2, 3}}]}
```

In[26]:=
```
Map[PermutationList[#, 4] &, D4A]
```

Out[26]=
```
{{1, 2, 3, 4}, {1, 3, 4, 2}, {1, 4, 2, 3}, {2, 1, 4, 3}, {2, 3, 1, 4}, {2, 4, 3, 1},
 {3, 1, 2, 4}, {3, 2, 4, 1}, {3, 4, 1, 2}, {4, 1, 3, 2}, {4, 2, 1, 3}, {4, 3, 2, 1}}
```

Sześcian/Ośmiościan foremny

!!! SymetricGroup - funkcja znajdująca grupę wszystkich bijekcji pewnego zbioru !!!

W zadaniu jest dane: Grupa jest izomorficzna z grupa(S4, ∘). Stąd wykorzystanie funkcji

In[24]:=
```
SymmetricGroup[4] // GroupOrder
```

Out[24]=
```
24
```

In[28]:=
```
D6S = SymmetricGroup[4] // GroupElements
```

Out[28]=
```
{Cycles[{}], Cycles[{{3, 4}}], Cycles[{{2, 3}}], Cycles[{{2, 3, 4}}], Cycles[{{2, 4, 3}}], Cycles[{{2, 4}}],
 Cycles[{{1, 2}}], Cycles[{{1, 2}, {3, 4}}], Cycles[{{1, 2, 3}}], Cycles[{{1, 2, 3, 4}}],
 Cycles[{{1, 2, 4, 3}}], Cycles[{{1, 2, 4}}], Cycles[{{1, 3, 2}}], Cycles[{{1, 3, 4, 2}}], Cycles[{{1, 3}}],
 Cycles[{{1, 3, 4}}], Cycles[{{1, 3}, {2, 4}}], Cycles[{{1, 3, 2, 4}}], Cycles[{{1, 4, 3, 2}}],
 Cycles[{{1, 4, 2}}], Cycles[{{1, 4, 3}}], Cycles[{{1, 4}}], Cycles[{{1, 4, 2, 3}}], Cycles[{{1, 4}, {2, 3}}]}
```

In[93]:=
```
Map[PermutationList[#, 4] &, D6S]
```

Out[93]=
```
{{1, 2, 3, 4}, {1, 2, 4, 3}, {1, 3, 2, 4}, {1, 3, 4, 2}, {1, 4, 2, 3}, {1, 4, 3, 2},
 {2, 1, 3, 4}, {2, 1, 4, 3}, {2, 3, 1, 4}, {2, 3, 4, 1}, {2, 4, 1, 3}, {2, 4, 3, 1},
 {3, 1, 2, 4}, {3, 1, 4, 2}, {3, 2, 1, 4}, {3, 2, 4, 1}, {3, 4, 1, 2}, {3, 4, 2, 1},
 {4, 1, 2, 3}, {4, 1, 3, 2}, {4, 2, 1, 3}, {4, 2, 3, 1}, {4, 3, 1, 2}, {4, 3, 2, 1}}
```

In[94]:=
```
Map[PermutationList[#, 4] &, D6S]
```

Out[94]=
```
{{1, 2, 3, 4}, {1, 2, 4, 3}, {1, 3, 2, 4}, {1, 3, 4, 2}, {1, 4, 2, 3}, {1, 4, 3, 2},
 {2, 1, 3, 4}, {2, 1, 4, 3}, {2, 3, 1, 4}, {2, 3, 4, 1}, {2, 4, 1, 3}, {2, 4, 3, 1},
 {3, 1, 2, 4}, {3, 1, 4, 2}, {3, 2, 1, 4}, {3, 2, 4, 1}, {3, 4, 1, 2}, {3, 4, 2, 1},
 {4, 1, 2, 3}, {4, 1, 3, 2}, {4, 2, 1, 3}, {4, 2, 3, 1}, {4, 3, 1, 2}, {4, 3, 2, 1}}
```

Sprawdzenie:

In[31]:=
```
Length[Map[PermutationList[#, 6] &, D6S]]
```

Out[31]=
```
24
```

In[32]:=
```
Length[Map[PermutationList[#, 8] &, D6S]]
```

Out[32]=
```
24
```

Dwudziestościan/Dwunastościan foremny

W zadaniu jest dane: Grupa jest izomorficzna z grupa$(A5, \circ)$.Stąd wykorzystanie funkcji A

In[33]:=
```
AlternatingGroup[5] // GroupOrder
```

Out[33]=
```
60
```

In[34]:=
```
D12A = AlternatingGroup[5] // GroupElements
```

Out[34]=
```
{Cycles[{}], Cycles[{{3, 4, 5}}], Cycles[{{3, 5, 4}}], Cycles[{{2, 3}, {4, 5}}], Cycles[{{2, 3, 4}}],
 Cycles[{{2, 3, 5}}], Cycles[{{2, 4, 3}}], Cycles[{{2, 4, 5}}], Cycles[{{2, 4}, {3, 5}}],
 Cycles[{{2, 5, 3}}], Cycles[{{2, 5, 4}}], Cycles[{{2, 5}, {3, 4}}], Cycles[{{1, 2}, {4, 5}}],
 Cycles[{{1, 2}, {3, 4}}], Cycles[{{1, 2}, {3, 5}}], Cycles[{{1, 2, 3}}], Cycles[{{1, 2, 3, 4, 5}}],
 Cycles[{{1, 2, 3, 5, 4}}], Cycles[{{1, 2, 4, 5, 3}}], Cycles[{{1, 2, 4}}], Cycles[{{1, 2, 4, 3, 5}}],
 Cycles[{{1, 2, 5, 4, 3}}], Cycles[{{1, 2, 5}}], Cycles[{{1, 2, 5, 3, 4}}], Cycles[{{1, 3, 2}}],
 Cycles[{{1, 3, 4, 5, 2}}], Cycles[{{1, 3, 5, 4, 2}}], Cycles[{{1, 3}, {4, 5}}], Cycles[{{1, 3, 4}}],
 Cycles[{{1, 3, 5}}], Cycles[{{1, 3}, {2, 4}}], Cycles[{{1, 3, 2, 4, 5}}], Cycles[{{1, 3, 5, 2, 4}}],
 Cycles[{{1, 3}, {2, 5}}], Cycles[{{1, 3, 2, 5, 4}}], Cycles[{{1, 3, 4, 2, 5}}],
 Cycles[{{1, 4, 5, 3, 2}}], Cycles[{{1, 4, 2}}], Cycles[{{1, 4, 3, 5, 2}}], Cycles[{{1, 4, 3}}],
 Cycles[{{1, 4, 5}}], Cycles[{{1, 4}, {3, 5}}], Cycles[{{1, 4, 5, 2, 3}}], Cycles[{{1, 4}, {2, 3}}],
 Cycles[{{1, 4, 2, 3, 5}}], Cycles[{{1, 4, 2, 5, 3}}], Cycles[{{1, 4, 3, 2, 5}}], Cycles[{{1, 4}, {2, 5}}],
 Cycles[{{1, 5, 4, 3, 2}}], Cycles[{{1, 5, 2}}], Cycles[{{1, 5, 3, 4, 2}}], Cycles[{{1, 5, 3}}],
 Cycles[{{1, 5, 4}}], Cycles[{{1, 5}, {3, 4}}], Cycles[{{1, 5, 4, 2, 3}}], Cycles[{{1, 5}, {2, 3}}],
 Cycles[{{1, 5, 2, 3, 4}}], Cycles[{{1, 5, 2, 4, 3}}], Cycles[{{1, 5, 3, 2, 4}}], Cycles[{{1, 5}, {2, 4}}]}
```

Dla 12kąta

In[67]:=
```
Map[PermutationList[#, 5] &, D12A]
```

Out[67]=
```
{{1, 2, 3, 4, 5}, {1, 2, 4, 5, 3}, {1, 2, 5, 3, 4}, {1, 3, 2, 5, 4}, {1, 3, 4, 2, 5}, {1, 3, 5, 4, 2},
 {1, 4, 2, 3, 5}, {1, 4, 3, 5, 2}, {1, 4, 5, 2, 3}, {1, 5, 2, 4, 3}, {1, 5, 3, 2, 4}, {1, 5, 4, 3, 2},
 {2, 1, 3, 5, 4}, {2, 1, 4, 3, 5}, {2, 1, 5, 4, 3}, {2, 3, 1, 4, 5}, {2, 3, 4, 5, 1}, {2, 3, 5, 1, 4},
 {2, 4, 1, 5, 3}, {2, 4, 3, 1, 5}, {2, 4, 5, 3, 1}, {2, 5, 1, 3, 4}, {2, 5, 3, 4, 1}, {2, 5, 4, 1, 3},
 {3, 1, 2, 4, 5}, {3, 1, 4, 5, 2}, {3, 1, 5, 2, 4}, {3, 2, 1, 5, 4}, {3, 2, 4, 1, 5},
 {3, 2, 5, 4, 1}, {3, 4, 1, 2, 5}, {3, 4, 2, 5, 1}, {3, 4, 5, 1, 2}, {3, 5, 1, 4, 2},
 {3, 5, 2, 1, 4}, {3, 5, 4, 2, 1}, {4, 1, 2, 5, 3}, {4, 1, 3, 2, 5}, {4, 1, 5, 3, 2},
 {4, 2, 1, 3, 5}, {4, 2, 3, 5, 1}, {4, 2, 5, 1, 3}, {4, 3, 1, 5, 2}, {4, 3, 2, 1, 5},
 {4, 3, 5, 2, 1}, {4, 5, 1, 2, 3}, {4, 5, 2, 3, 1}, {4, 5, 3, 1, 2}, {5, 1, 2, 3, 4},
 {5, 1, 3, 4, 2}, {5, 1, 4, 2, 3}, {5, 2, 1, 4, 3}, {5, 2, 3, 1, 4}, {5, 2, 4, 3, 1},
 {5, 3, 1, 2, 4}, {5, 3, 2, 4, 1}, {5, 3, 4, 1, 2}, {5, 4, 1, 3, 2}, {5, 4, 2, 1, 3}, {5, 4, 3, 2, 1}}
```

Dla 20kąta

In[68]:=
```
Map[PermutationList[#, 5] &, D12A]
```

Out[68]=
```
{{1, 2, 3, 4, 5}, {1, 2, 4, 5, 3}, {1, 2, 5, 3, 4}, {1, 3, 2, 5, 4}, {1, 3, 4, 2, 5}, {1, 3, 5, 4, 2},
 {1, 4, 2, 3, 5}, {1, 4, 3, 5, 2}, {1, 4, 5, 2, 3}, {1, 5, 2, 4, 3}, {1, 5, 3, 2, 4}, {1, 5, 4, 3, 2},
 {2, 1, 3, 5, 4}, {2, 1, 4, 3, 5}, {2, 1, 5, 4, 3}, {2, 3, 1, 4, 5}, {2, 3, 4, 5, 1}, {2, 3, 5, 1, 4},
 {2, 4, 1, 5, 3}, {2, 4, 3, 1, 5}, {2, 4, 5, 3, 1}, {2, 5, 1, 3, 4}, {2, 5, 3, 4, 1}, {2, 5, 4, 1, 3},
 {3, 1, 2, 4, 5}, {3, 1, 4, 5, 2}, {3, 1, 5, 2, 4}, {3, 2, 1, 5, 4}, {3, 2, 4, 1, 5},
 {3, 2, 5, 4, 1}, {3, 4, 1, 2, 5}, {3, 4, 2, 5, 1}, {3, 4, 5, 1, 2}, {3, 5, 1, 4, 2},
 {3, 5, 2, 1, 4}, {3, 5, 4, 2, 1}, {4, 1, 2, 5, 3}, {4, 1, 3, 2, 5}, {4, 1, 5, 3, 2},
 {4, 2, 1, 3, 5}, {4, 2, 3, 5, 1}, {4, 2, 5, 1, 3}, {4, 3, 1, 5, 2}, {4, 3, 2, 1, 5},
 {4, 3, 5, 2, 1}, {4, 5, 1, 2, 3}, {4, 5, 2, 3, 1}, {4, 5, 3, 1, 2}, {5, 1, 2, 3, 4},
 {5, 1, 3, 4, 2}, {5, 1, 4, 2, 3}, {5, 2, 1, 4, 3}, {5, 2, 3, 1, 4}, {5, 2, 4, 3, 1},
 {5, 3, 1, 2, 4}, {5, 3, 2, 4, 1}, {5, 3, 4, 1, 2}, {5, 4, 1, 3, 2}, {5, 4, 2, 1, 3}, {5, 4, 3, 2, 1}}
```

Sprawdzenie:

In[37]:=
```
Length[Map[PermutationList[#, 12] &, D12A]]
```

Out[37]=
```
60
```

In[42]:=
```
Length[Map[PermutationList[#, 20] &, D12A]]
```

Out[42]=
```
60
```

W zadaniach c,d oraz e wykorzystana jest własność jaką jest izomorfia grup. Podane są informacje o izomofriach poszczególnych grup

(A4 - grupa alternująca dla 4)

(S4 - grupa symetryczna dla 4 (2 * A4))

(A5 - grupa alternująca dla 5)

dzięki temu w prosty sposób można wykorzystać te funkcję ponieważ izomorfia gwarantuje nam poprawność wyników

In[107]:=

```
(* Funkcja klasyfikująca permutacje w grupie symetrii *)
ClassifyPermutation[perm_Cycles] := Module[{cycles, lengths},
 cycles = PermutationCycles[perm];
 lengths = Length[cycles];
 Which[
  perm == Cycles[{}], "Identyczność",
  TrueQ[lengths == 4], "Obrót przez środek bryły",
 TrueQ[lengths == 3], "Obrót przez środek ściany",
  TrueQ[Length[cycles] == 2], "Obrót przez środek krawędzi",
  True, "Inna izometria"
 ]
];



A4 = GroupElements[AlternatingGroup[4]];
classifiedPermutations = Table[{el, ClassifyPermutation[el]}, {el, A4}];
classifiedPermutations // Grid
(* jest błąd w funklcji sprawdzającej,
sprawdza długość cycles(czyli ile jest a nie długość cyklu,
   jak się da można naprawić żeby liczyło długość cyklu konkretnego i będzie działać*)
```

Out[110]=

Cycles[{}]	Identyczność
Cycles[{{2, 3, 4}}]	Inna izometria
Cycles[{{2, 4, 3}}]	Inna izometria
Cycles[{{1, 2}, {3, 4}}]	Inna izometria
Cycles[{{1, 2, 3}}]	Inna izometria
Cycles[{{1, 2, 4}}]	Inna izometria
Cycles[{{1, 3, 2}}]	Inna izometria
Cycles[{{1, 3, 4}}]	Inna izometria
Cycles[{{1, 3}, {2, 4}}]	Inna izometria
Cycles[{{1, 4, 2}}]	Inna izometria
Cycles[{{1, 4, 3}}]	Inna izometria
Cycles[{{1, 4}, {2, 3}}]	Inna izometria