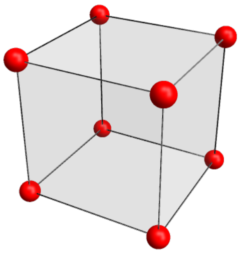
```
In[1]:= (*Numerujemy wierzchołki dokładnie tak,jak w RysujSzescian:1↦{-1,-1,-1},
2↦{1,-1,-1},3↦{1,1,-1},4↦{-1,1,-1},5↦{-1,-1,1},6↦{1,-1,1},7↦{1,1,1},
8↦{-1,1,1}.*)(* ===1. Poprawne generatory dla obrotów o 90° wokół osi Z,
Y i X===*)(*Obrót o 90° wokół osi Z (patrząc od+Z w dół,czyli z góry-
przeciwnie do ruchu wskazówek zegara).(x,y,z)→(-y,x,z).W efekcie:1=
{-1,-1,-1}→(-(-1), -1, -1)={1, -1, -1} czyli 2 2={1,-1,-1}→(-(-1),1, -1)=
{1,1, -1} czyli 3 3={1,1,-1}→(-1,1, -1)={-1,1, -1} czyli 4 4=
{-1,1,-1}→(-1, -1, -1)={-1, -1, -1} czyli 1 Analogicznie wierzchołki 5-
8 (z=+1):(5→6→7→8→5).*)obrótZ = Cycles[{{1, 2, 3, 4}, {5, 6, 7, 8}}];
cykle

(*Obrót o 90° wokół osi Y (patrząc od+Y w dół,czyli od przodu-
przeciwnie do ruchu wskazówek zegara).Równanie:(x,y,z)→(z,y, -x).Sprawdźmy,
co dzieje się z wierzchołkami:1={-1,-1,-1}→(-1, -1, -(-1))={-1, -1,1} czyli 5 5=
{-1,-1,1}→(1, -1, -(-1))={1, -1, -1} czyli 2 2={1,-1,-1}→(-1, -1, -1)=
{-1, -1, -1} czyli 1 6={1,-1,1}→(1, -1, -(1))={1, -1, -1} czyli 2<
-uwaga:Ale żeby zachować porządek cyklu po 4 elementy,
rozbijmy to w pary dwulistowe:Po kolei dla całej „dolnej” warstwy
i „górnej”:Dolna warstwa (z= -1):
1→5→6→2→1?Ale zobaczmy dokładnie:1(-1,-1,-1)→5(-1,-1,1) 5(-1,-1,1)→
6(1,-1,1) 6(1,-1,1)→2(1,-1,-1) 2(1,-1,-1)→1(-1,-1,-1) Cykl w dolnej połowie:
(1,5,6,2).Górna warstwa (z=+1,y=+1):4(-1,1,-1)→8(-1,1,1)→
7(1,1,1)→3(1,1,-1)→4 Cykl w górnej połowie:(4,8,7,3).*)
obrótY = Cycles[{{1, 5, 6, 2}, {4, 8, 7, 3}}];
cykle

(*Obrót o 90° wokół osi X
(patrząc od+X w dół,czyli od prawej strony sześcianu-przeciwnie do ruchu
wskazówek zegara).Równanie:(x,y,z)→(x, -z,y).Dla wierzchołków:1=
{-1, -1, -1}→(-1, -(-1), -1)={-1,1, -1} czyli 4 4={-1,1, -1}→(-1, -(-1),1)=
{-1,1,1} czyli 8 8={-1,1,1}→(-1, -(1),1)={-1, -1,1} czyli 5 5=
{-1, -1,1}→(-1, -(1),  -1)={-1, -1, -1} czyli 1 Cykl przy x= -
1:(1,4,8,5).Dolna warstwa (x=+1):2={1, -1, -1}→(1, -(-1), -1)=
{1,1, -1} czyli 3 3={1,1, -1}→(1, -(-1),1)=
{1,1,1} czyli 7 7={1,1,1}→(1, -(1),1)={1, -1,1} czyli 6 6=
{1, -1,1}→(1, -(1),  -1)={1, -1, -1} czyli 2 Cykl przy x=+1:(2,3,7,6).*)
obrótX = Cycles[{{1, 4, 8, 5}, {2, 3, 7, 6}}];
cykle

(*Utworzenie grupy z tych trzech generatorów*)
grupa = PermutationGroup[{obrótZ, obrótY, obrótX}];
grupa permutacji
elementyGrupy = GroupElements[grupa];
elementy grupy

(* ===2. Wszystkie możliwe kolorowania 8 wierzchołków dwiema barwami (0 i 1)===*)
wszystkieKolorowania = Tuples[{0, 1}, 8];
krotki

(*Słowniki pomocnicze do wyszukiwania orbity*)
odwiedzoneWszystkie = <||>; (*będzie trzymać już odwiedzone wektory*)
```

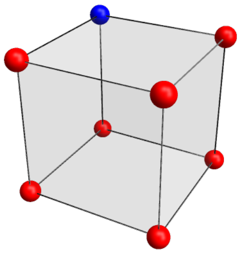
```
reprezentanci = {};     (*tu zebierzemy ostatecznych reprezentantów orbity*)

(* ===3. Wyznaczanie reprezentantów orbity metodą BFS===*)
Do[kolorowanie = wszystkieKolorowania[[i]];
rób
  (*Jeżeli to kolorowanie nie było jeszcze w żadnej orbicie...*)
  If[! KeyExistsQ[odwiedzoneWszystkie, kolorowanie], (*Tworzymy pusty słownik,
  op⋯ istnieje klucz?
   który w miarę BFS wypełni wszystkie elementy orbity*)orbita = <| |>;
   kolejka = {kolorowanie};
   (*Klasyczne BFS:dla każdego obecnego wektora generujemy wszystkie obrazy przez
      każdy element grupy*)While[kolejka =!= {}, obecne = First[kolejka];
                           podczas                          pierwszy
    kolejka = Rest[kolejka];
               bez pierwszego elementu
    If[! KeyExistsQ[orbita, obecne], AssociateTo[orbita, obecne → True];
    op⋯ istnieje klucz?            dodaj to stowarzyszenia      prawda
     (*Dla każdej permutacji g z grupy generujemy nowe wektory*)
     Do[nowe = Permute[obecne, InversePermutation[elementyGrupy[[j]]]];
     rób       permutuj         permutacja odwrotna
      If[! KeyExistsQ[orbita, nowe], AppendTo[kolejka, nowe];];,
      op⋯ istnieje klucz?          dołącz na końcu do wartości zmiennej
      {j, Length[elementyGrupy]}];];];
          długość
   (*Po zakończonym BFS wszystkie klucze w orbita
    to wektory z jednej orbity."kolorowanie" to pierwszy wektor,
   dodajemy go jako reprezentanta.*)AppendTo[reprezentanci, kolorowanie];
                                    dołącz na końcu do wartości zmiennej
   (*Oznaczamy każdy wektor z tej orbity jako odwiedzony,
   aby już go nie rozpatrywać ponownie*)
   Do[AssociateTo[odwiedzoneWszystkie, c → True], {c, Keys[orbita]}];];,
   rób dodaj to stowarzyszenia            prawda    klucze
  {i, Length[wszystkieKolorowania]}];
      długość

(* ===4. Wyświetlenie liczby
  nieekwiwalentnych kolorowań oraz samych reprezentantów===*)
Print["Liczba nieekwiwalentnych kolorowań: ", Length[reprezentanci]];
drukuj                                        długość

Print["Reprezentanci orbity (posortowani
drukuj
     według liczby jedynek = liczby wierzchołków w kolorze 1):"];
posortowaniReprezentanci = SortBy[reprezentanci, Total];
                           sortuj według         oblicz sumę

Do[Print[posortowaniReprezentanci[[k]]], {k, Length[posortowaniReprezentanci]}];
rób drukuj                                   długość

(* ===5. Funkcja rysująca sześcian z pokolorowanymi wierzchołkami===*)
RysujSzescian[kolorowanie_] := Module[
                               moduł
```

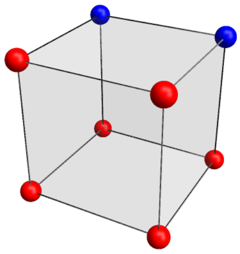
```
{wierzcholki3D, sciany, krawedzieIndeksy, scianyPolygon, krawedzieLines, kolory},
(*3D-współrzędne ośmiu wierzchołków*)wierzcholki3D = {{-1, -1, -1}, {1, -1, -1},
  oblicz pochodną
  {1, 1, -1}, {-1, 1, -1}, {-1, -1, 1}, {1, -1, 1}, {1, 1, 1}, {-1, 1, 1}};
(*Ściany sześcianu (każda jako czwórka indeksów wierzchołków)*)
sciany = {{1, 2, 3, 4}, (*dolna*){5, 6, 7, 8}, (*górna*){1, 2, 6, 5}, (*przód*)
  {2, 3, 7, 6}, (*prawa*){3, 4, 8, 7}, (*tył*){4, 1, 5, 8} (*lewa*)};
(*Krawędzie:para indeksów wierzchołków*)krawedzieIndeksy = {{1, 2}, {2, 3}, {3, 4},
  {4, 1}, {5, 6}, {6, 7}, {7, 8}, {8, 5}, {1, 5}, {2, 6}, {3, 7}, {4, 8}};
(*Tworzymy obiekty graficzne dla ścian i krawędzi*)
scianyPolygon = Polygon /@ (wierzcholki3D[[#]] & /@ sciany);
                wielokąt
krawedzieLines = Line /@ (wierzcholki3D[[#]] & /@ krawedzieIndeksy);
                 linia łamana
(*Zamiana 0→Red,1→Blue*)kolory = kolorowanie /. {0 → Red, 1 → Blue};
          czer···  niebieski                    czerwony  niebieski
Graphics3D[{{Opacity[0.1], Gray, scianyPolygon},
trójwymiarowa g··  nieprzezroczystość  szary
  (*przezroczyste ściany*){Gray, Thickness[0.005], krawedzieLines},
                           szary  grubość
  (*krawędzie*)(*wierzchołki jako barwione kulki*)MapThread[
                                                  zastosuj w wątku
   {#2, Specularity[White, 50], Sphere[#1, 0.15]} &, {wierzcholki3D, kolory}]},
        odblaskowość  biały     sfera
 Boxed → False, Lighting → "Neutral", ViewPoint → {1.3, -2.4, 1.5},
 dodaj ···  fałsz  oświetlenie            punkt widzenia
 ImageSize → 150]];
 rozmiar obrazu

(* ===6. Generowanie i wyświetlanie wizualizacji===*)
wizualizacje = Table[Labeled[RysujSzescian[posortowaniReprezentanci[[i]]],
               tabela  z etykietą
   Row[{"Kolorowanie ", i}], Bottom], {i, Length[posortowaniReprezentanci]}];
   rząd                     dół         długość

siatka = Partition[wizualizacje, UpTo[5]];
         podział na rozdzielne bloki  aż do

Print[Grid[siatka, Frame → All, FrameStyle → LightGray]];
drukuj krata   ramka  ws···  styl ramki  jasnoszary
```

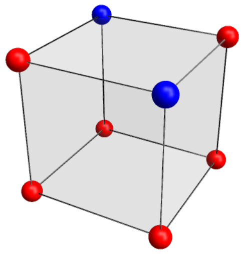
```
Liczba nieekwiwalentnych kolorowań: 23

Reprezentanci orbity (posortowani według liczby jedynek = liczby wierzchołków w kolorze 1):

{0, 0, 0, 0, 0, 0, 0, 0}

{0, 0, 0, 0, 0, 0, 0, 1}

{0, 0, 0, 0, 0, 0, 1, 1}

{0, 0, 0, 0, 0, 1, 0, 1}

{0, 0, 0, 1, 0, 1, 0, 0}

{0, 0, 0, 0, 0, 1, 1, 1}
```

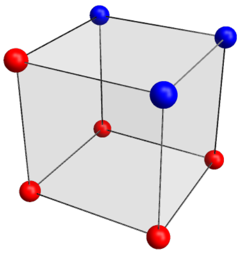
```
{0, 0, 0, 1, 0, 1, 0, 1}
{0, 0, 0, 1, 1, 0, 1, 0}
{0, 0, 0, 0, 1, 1, 1, 1}
{0, 0, 0, 1, 0, 1, 1, 1}
{0, 0, 0, 1, 1, 0, 1, 1}
{0, 0, 0, 1, 1, 1, 0, 1}
{0, 0, 0, 1, 1, 1, 1, 0}
{0, 0, 1, 1, 1, 1, 0, 0}
{0, 1, 0, 1, 1, 0, 1, 0}
{0, 0, 0, 1, 1, 1, 1, 1}
{0, 0, 1, 1, 1, 1, 0, 1}
{0, 1, 0, 1, 1, 0, 1, 1}
{0, 0, 1, 1, 1, 1, 1, 1}
{0, 1, 0, 1, 1, 1, 1, 1}
{0, 1, 1, 1, 1, 1, 0, 1}
{0, 1, 1, 1, 1, 1, 1, 1}
{1, 1, 1, 1, 1, 1, 1, 1}
```

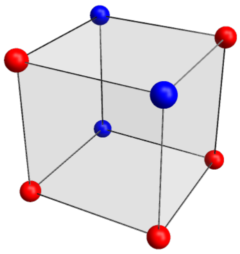

Kolorowanie 1

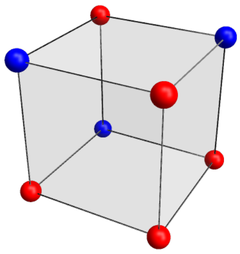

Kolorowanie 2

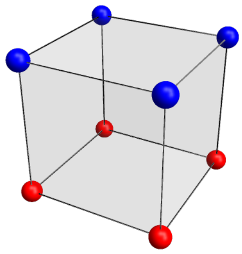

Kolorowanie 3

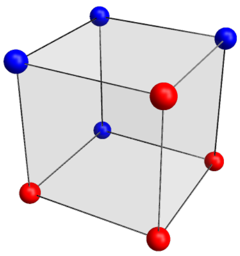

Kolorowanie 4

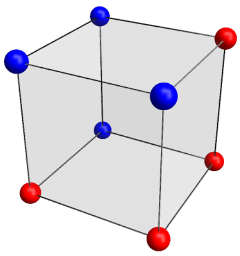

Kolorowanie 6

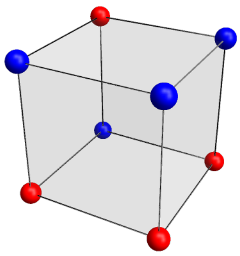

Kolorowanie 7

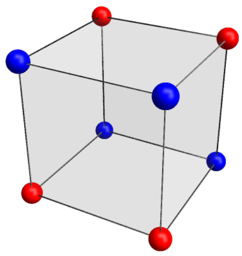

Kolorowanie 8

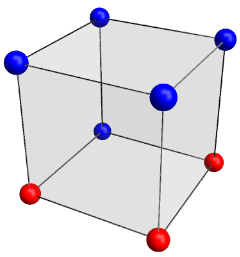

Kolorowanie 9

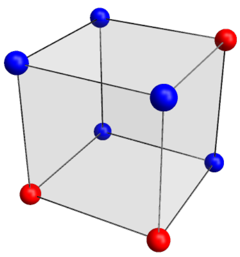

Kolorowanie 11

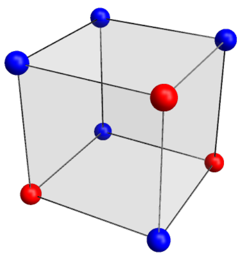

Kolorowanie 12

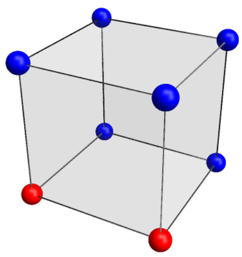

Kolorowanie 13

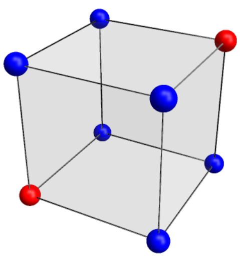

Kolorowanie 14

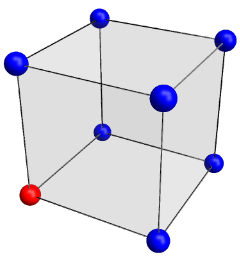

Kolorowanie 16

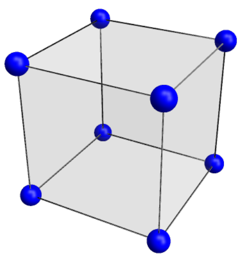

Kolorowanie 17


Kolorowanie 18


Kolorowanie 19


Kolorowanie 21


Kolorowanie 22


Kolorowanie 23