Zajęcia III : Ergodyczność

Zad. 1(Zastosowanie Tw. Birkhoffa)

Oczywistym jest, że obrót na okręgu zachowuje miarę lebesgue' a.

Co więcej z zajęć I możemy wnioskować, ze obrót o kąt niewspółmierny do π jest przekształceniem ergodycznym.

I. definujemy funkcję zwracającą orbitę długości n startującą z punktu $x_0 \in (0, 2\pi)$ w funkcji obrotu o kąt ϕ.

```
orbitaobrotu[x0_, ϕ_, n_] := NestList[N[Mod[# + ϕ, 2 Pi]] &, x0, n - 1]
```

Losujemy jeszcze przykładowe warości x_0 oraz ϕ :

```
x0 = RandomReal[{0, 2 Pi}];
ϕ = RandomReal[{0, 2 Pi}];
```

II. Za pomocą funkcji Map[] będziemy stosować zadaną funcję do każdej współrzędnej wygenerowanej orbity.:

Przykładowo :

```
Map[ Cos, orbitaobrotu[x0, ϕ, 5]]
{0.388754, 0.697502, -0.931495, 0.0273125, 0.910243}
```

Zapisujemy funkcję wyliczającą sumy z twierdzenia Birkhoffa pomnożone przez $\lambda([0, 2\pi]) = 2\pi$

```
aproksymacja[fun_, x0_, ϕ_, n_] := 2 Pi Total[Map[fun, orbitaobrotu[x0, ϕ, n]]] / n
```

II Obliczamy zadane całki

a) $\int_0^{2\pi} \sin(x)\, dx$

Wartość dokładna

```
Integrate[Sin[x], {x, 0, 2 Pi}]
0
```

wartości przybliżone

```
Table[a = aproksymacja[Sin, x0, ϕ, 10^i];
 "Dla n=" <> ToString[Superscript[10, ToString[i]], StandardForm] <>
  " wartość przybliżona to: " <> ToString[a, TraditionalForm] <>
  " błąd bezwzględny wynosi:" <> ToString[N[Abs[a]], TraditionalForm], {i, 0, 4}]
```

{Dla n=10^0 wartość przybliżona to: 5.78896 błąd bezwzględny wynosi:5.78896,
Dla n=10^1 wartość przybliżona to: 0.30849 błąd bezwzględny wynosi:0.30849,
Dla n=10^2 wartość przybliżona to: 0.0583333 błąd bezwzględny wynosi:0.0583333,
Dla n=10^3 wartość przybliżona to: 0.00192336 błąd bezwzględny wynosi:0.00192336,
Dla n=10^4 wartość przybliżona to: 1.16581×10^{-6} błąd bezwzględny wynosi:1.16581×10^{-6}}

a) $\int_0^{2\pi} \cos^2(x)\, dx$

```
f2[x_] := (Cos[x])^2
```

Wartość dokładna

```
N[Integrate[f2[x], {x, 0, 2 Pi}]]
```

3.14159

wartości przybliżone

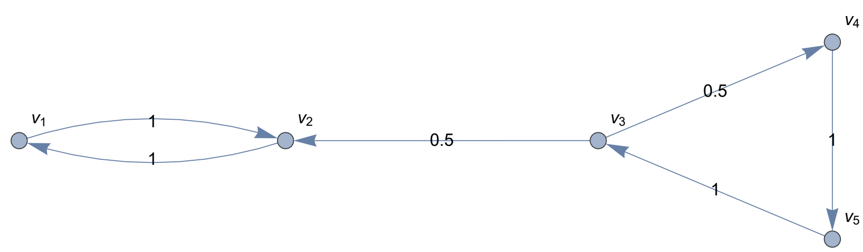
```
Table[a = aproksymacja[f2, x0, ϕ, 10^i];
 "Dla n=" <> ToString[Superscript[10, ToString[i]], StandardForm] <>
  " wartość przybliżona to: " <> ToString[a, TraditionalForm] <>
  " błąd bezwzględny wynosi:" <> ToString[N[Abs[a - Pi]], TraditionalForm], {i, 0, 4}]
```

{Dla n=10^0 wartość przybliżona to: 0.949578 błąd bezwzględny wynosi:2.19201,
Dla n=10^1 wartość przybliżona to: 2.89721 błąd bezwzględny wynosi:0.244379,
Dla n=10^2 wartość przybliżona to: 3.12944 błąd bezwzględny wynosi:0.0121491,
Dla n=10^3 wartość przybliżona to: 3.13988 błąd bezwzględny wynosi:0.0017169,
Dla n=10^4 wartość przybliżona to: 3.14159 błąd bezwzględny wynosi:4.95409×10^{-7}}

Zad. II (Przykłady Łańcuchów Markowa)

a) łańcuch nieokresowy:

```
a1 = {{∞, 1, ∞, ∞, ∞},
   {1, ∞, 0.5, ∞, ∞},
   {∞, ∞, ∞, ∞, 1},
   {∞, ∞, 0.5, ∞, ∞},
   {∞, ∞, ∞, 1, ∞}};
WeightedAdjacencyGraph[Table[v_i, {i, 5}], Transpose[a1],
 VertexLabels → "Name", EdgeLabels → "EdgeWeight", ImagePadding → 10]
"Macierz Przejścia w jednym kroku:" MatrixForm[a1 /. {∞ → 0} ]
```



Macierz Przejścia w jednym kroku: $\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

Okresy stanów :

o(1) = NWD ({0,2,4,6,...}) = 2 - stan o okresie 2,

o(2) = NWD ({0,2,4,6,...}) = 2 - stan o okresie 2,

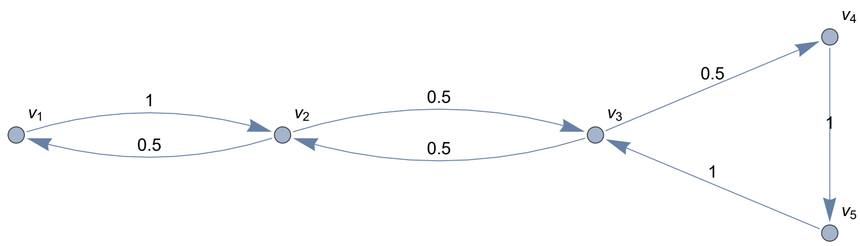
o(3) = NWD ({3,6,9,...}) = 3 - stan o okresie 3,

o(4) = NWD ({3,6,9,...}) = 3 - stan o okresie 3,

o(5) = NWD ({3,6,9,...}) = 3 - stan o okresie 3.

b) łańcuch nieprzewiedlny (wszystkie stany się komunikują):

```
a2 = {{∞, 0.5, ∞, ∞, ∞},
    {1, ∞, 0.5, ∞, ∞},
    {∞, 0.5, ∞, ∞, 1},
    {∞, ∞, 0.5, ∞, ∞},
    {∞, ∞, ∞, 1, ∞}};
WeightedAdjacencyGraph[Table[v_i, {i, 5}], Transpose[a2],
 VertexLabels → "Name", EdgeLabels → "EdgeWeight", ImagePadding → 10]
"Macierz Przejścia w jednym kroku:" MatrixForm[a2 /. {∞ → 0} ]
```



Macierz Przejścia w jednym kroku: $\begin{pmatrix} 0 & 0.5 & 0 & 0 & 0 \\ 1 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 1 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

Okresy stanów :

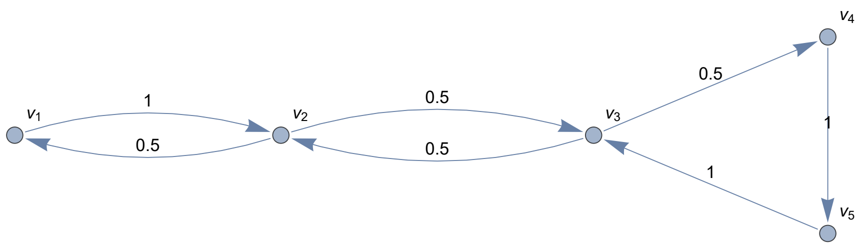
o(1) = NWD ({0,2,4,6,7,...}) = 1 - stan nieokresowy,

Z Twierdzenia: W nieprzewiedlnym łancuchu Markowa wszystkie stany mają ten sam okres.
Wiemy, że każdy punkt jest nieokresowy i o(1)=o(2)=o(3)=o(4)=o(5)=1. Stąd łancuch jest nieokresowy.

c) łańcuch ergodyczny:

Na mocy twierdzenia : Każdy nieokresowy i nieprzewiedlny łancuch Markowa jest ergodyczny możemy użyć przykładu b).

```
a3 = {{∞, 0.5, ∞, ∞, ∞},
    {1, ∞, 0.5, ∞, ∞},
    {∞, 0.5, ∞, ∞, 1},
    {∞, ∞, 0.5, ∞, ∞},
    {∞, ∞, ∞, 1, ∞}};
WeightedAdjacencyGraph[Table[v_i, {i, 5}], Transpose[a3],
 VertexLabels → "Name", EdgeLabels → "EdgeWeight", ImagePadding → 10]
"Macierz Przejścia w jednym kroku:" MatrixForm[a3 /. {∞ → 0} ]
```



Macierz Przejścia w jednym kroku: $\begin{pmatrix} 0 & 0.5 & 0 & 0 & 0 \\ 1 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 1 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

Zad. III (Szukanie rozkładów stacjonarnych)

a) Wykorzystując własnosci macierzy przejścia (nie ważne czy łańcuch jest ergodyczny).

Rozkład stacjonany (w tym przypadku ergodyczny) możemy znaleźć wykorzystując fakt, że jest on wektorem własnym dla wartości własnej równej 1.

```
Eigensystem[a3 /. {∞ → 0}]
```

```
{{1., -0.790018, -0.395706 + 0.622637 i, -0.395706 - 0.622637 i, 0.581429},
 {{-0.301511, -0.603023, -0.603023, -0.301511, -0.301511},
  {-0.5, 0.790018, -0.248256, 0.157121, -0.198882}, {0.146995 - 0.0990756 i,
   0.00704276 + 0.261459 i, -0.625151 + 0. i, 0.227259 + 0.357589 i, 0.243855 - 0.519972 i},
  {0.146995 + 0.0990756 i, 0.00704276 - 0.261459 i, -0.625151 + 0. i, 0.227259 - 0.357589 i,
   0.243855 + 0.519972 i}, {0.5, 0.581429, -0.32388, -0.278521, -0.479028}}}
```

Jest tylko jedna wartość własna równą 1 stąd rozkad stacjonarny jest jednoznaczny.
 wektor własny dla tej wartości własnej to :

```
Re[Eigensystem[a3 /. {∞ → 0}][[2]][[1]]]
```

```
{-0.301511, -0.603023, -0.603023, -0.301511, -0.301511}
```

Stąd mamy następujący rozkład stacjonarny :

```
rozkladstacjonarny3 = Re[Eigensystem[a3 /. {∞ → 0}][[2]][[1]]] /
  Total[Re[Eigensystem[a3 /. {∞ → 0}][[2]][[1]]]]
```

{0.142857, 0.285714, 0.285714, 0.142857, 0.142857}

Sprawdzenie :

```
aa3 = a3 /. {∞ → 0};
aa3.rozkladstacjonarny3 == rozkladstacjonarny3
```

True

b) Dla procesów ergodycznych możemy uzyskać rozkład stacjonarny poprzez iterację macierzy przejscia.

```
MatrixPower[aa3, 1] // MatrixForm
MatrixPower[aa3, 10] // MatrixForm
MatrixPower[aa3, 100] // MatrixForm
MatrixPower[aa3, 1000] // MatrixForm
```

$$\begin{pmatrix} 0. & 0.5 & 0. & 0. & 0. \\ 1. & 0. & 0.5 & 0. & 0. \\ 0. & 0.5 & 0. & 0. & 1. \\ 0. & 0. & 0.5 & 0. & 0. \\ 0. & 0. & 0. & 1. & 0. \end{pmatrix}$$

$$\begin{pmatrix} 0.189453 & 0.105469 & 0.157227 & 0.167969 & 0.117188 \\ 0.210938 & 0.34668 & 0.273438 & 0.234375 & 0.314453 \\ 0.314453 & 0.273438 & 0.266602 & 0.292969 & 0.3125 \\ 0.117188 & 0.157227 & 0.15625 & 0.109375 & 0.146484 \\ 0.167969 & 0.117188 & 0.146484 & 0.195313 & 0.109375 \end{pmatrix}$$

$$\begin{pmatrix} 0.142857 & 0.142857 & 0.142857 & 0.142857 & 0.142857 \\ 0.285714 & 0.285714 & 0.285714 & 0.285714 & 0.285714 \\ 0.285714 & 0.285714 & 0.285714 & 0.285714 & 0.285714 \\ 0.142857 & 0.142857 & 0.142857 & 0.142857 & 0.142857 \\ 0.142857 & 0.142857 & 0.142857 & 0.142857 & 0.142857 \end{pmatrix}$$

$$\begin{pmatrix} 0.142857 & 0.142857 & 0.142857 & 0.142857 & 0.142857 \\ 0.285714 & 0.285714 & 0.285714 & 0.285714 & 0.285714 \\ 0.285714 & 0.285714 & 0.285714 & 0.285714 & 0.285714 \\ 0.142857 & 0.142857 & 0.142857 & 0.142857 & 0.142857 \\ 0.142857 & 0.142857 & 0.142857 & 0.142857 & 0.142857 \end{pmatrix}$$

Macierz przejścia "ustabilizowała się" dla dostatecznie dużej iteracji stąd wnioskujemy, że rozkład ergodyczny dany jest przez

```
rozkladergodyczny3 = MatrixPower[aa3, 1000][[ ;; , 1]]
```

{0.142857, 0.285714, 0.285714, 0.142857, 0.142857}

Sprawdzenie Ergodyczności:

1) Czy sumuje się do jedności :

```
Total[rozkladergodyczny3]
```

1.

2) Czy wszystkie elementy rozkladu stacjonarnego są dodatnie

```
Positive[rozkladergodyczny3] == ConstantArray[True, Length[rozkladergodyczny3]]
```

```
True
```

3) Czy jest to wektor wlasny dla wartości własnej równej 1

```
aa3.rozkladergodyczny3 == rozkladergodyczny3
```

```
True
```

Dodatkowo widzimy, że wektor uzyskany w ten sposób jest równy uzyskanemu w podpunkcie a)

```
rozkladergodyczny3 == rozkladstacjonarny3
```

```
True
```

Ciekawostka do zadania III - automatyzacja znajdująca wszystkie rozkłady stacjonarne dla zadanej macierzy przejścia.

```
automatyzacja[m_] :=
 {
   pom1 = Eigensystem[m];
   zero = ConstantArray[0, Length[pom1[[1]]]];
   (* pom2 to juz właściwie jest wynik,
   jesgo celem jest zwrócenie listy wektorów własnych dla wartości własnych 1
    (może być ich więcej niż jedna jeżeli nie będzie to łańcuch ergodyczny *)
   pom2 = Nest[
      {
        #[[1]] + 1,
        If[
         Element[pom1[[1]][[#[[1]]]], Reals],
         If[
          Round[pom1[[1]][[#[[1]]]], 0.0001] == 1,
          Append[#[[2]], pom1[[2]][[#[[1]]]]],
          #[[2]]
         ],
         #[[2]]
        ]
       } &,
      {1, {}},
      Length[pom1[[1]]]
     ][[2]];
   (* Ten Table[] poniżej jest po to aby pozbyć się
    wartości 0.i w wektorach własnych dodatkowo od razu je normujemy *)
   pom3 = Table[
     If[Im[pom2[[k]]] == zero,
      Re[pom2[[k]]] / Total[Re[pom2[[k]]]], "wetor własny był zespolony !!"],
     {k, 1, Length[pom2]}
    ];
   (* Część poniżej służy do ładnego wyświetlania wyniku *)
   "Łańcuch Markowa dany macierzą przejścia " <>
    ToString[MatrixForm[m] , StandardForm] <>

    "
posiada " <> ToString[Length[pom3]] <> If[Length[pom3] == 1, " rozkład stacjonarny: ",
     If[Length[pom3] < 5, " rozkłady stacjonarne: ", " rozkładów stacjonarnych: "]]
    <> "
" <> Table[ToString[i] <> ". " <> ToString[pom3[[i]]] <> If[i == Length[pom3], ".", ",
"], {i, 1, Length[pom3]}]
  }[[1]]
```

Przykłady :

1. Macierz przejścia ta samo co wcześńiej dla łańcucha ergodycznego:

```
automatyzacja[a3 /. {∞ → 0}]
```

łańcuch Markowa dany macierzą przejścia $\begin{pmatrix} 0 & 0.5 & 0 & 0 & 0 \\ 1 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 1 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

posiada 1 rozkład stacjonarny:
1. {0.142857, 0.285714, 0.285714, 0.142857, 0.142857}.

2. Macierz przejścia dla łańcucha nieergodycznego:

```
a7 = {{0.2, 0.5, 0.4, 0, 0.3, 0, 0},
    {0.8, 0.5, 0.1, 0.2, 0, 1, 0},
    {0, 0, 0.2, 0.4, 0.2, 0, 0},
    {0, 0, 3, 0.4, 0.3, 0, 0},
    {0, 0, 0, 0, 0.2, 0, 0},
    {0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 1}};
```

```
automatyzacja[a7]
```

łańcuch Markowa dany macierzą przejścia $\begin{pmatrix} 0.2 & 0.5 & 0.4 & 0 & 0.3 & 0 & 0 \\ 0.8 & 0.5 & 0.1 & 0.2 & 0 & 1 & 0 \\ 0 & 0 & 0.2 & 0.4 & 0.2 & 0 & 0 \\ 0 & 0 & 3 & 0.4 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

posiada 2 rozkłady stacjonarne:
1. {0.384615, 0.615385, 0., 0., 0., 0., 0.},
2. {0., 0., 0., 0., 0., 0., 1.}.

```
automatyzacja[IdentityMatrix[4]]
```

łańcuch Markowa dany macierzą przejścia $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

posiada 4 rozkłady stacjonarne:
1. {0, 0, 0, 1},
2. {0, 0, 1, 0},
3. {0, 1, 0, 0},
4. {1, 0, 0, 0}.

Zad. IV (teoretyczny szczur w teoretycznym labiryncie)

Zapisujemy macierz przejścia (w konwencji podanej na zajęciach) dla szczura :

```
p = {{0, 1/3, 0, 1/3, 0, 0, 0, 0, 0},
   {1/2, 0, 0, 0, 1/4, 0, 0, 0, 0},
   {0, 1/3, 1, 0, 0, 1/3, 0, 0, 0},
   {1/2, 0, 0, 0, 1/4, 0, 0, 0, 0},
   {0, 1/3, 0, 1/3, 0, 1/3, 0, 1/3, 0},
   {0, 0, 0, 0, 1/4, 0, 0, 0, 1/2},
   {0, 0, 0, 1/3, 0, 0, 1, 1/3, 0},
   {0, 0, 0, 0, 1/4, 0, 0, 0, 1/2},
   {0, 0, 0, 0, 0, 1/3, 0, 1/3, 0}};
```

Sprawdzenie

```
Total[p]
```

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

```
MatrixForm[p]
```

$$\begin{pmatrix} 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 1 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 1 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \end{pmatrix}$$

Szukamy prawdopodobienstw osiagniecia stanów pochłaniających :

Zauważamy, że z symetrii labiryntu prawdopodobnieństwa osiągnięcia sera i kota są równe (1/2) dla komnat : 1, 5, 6
startując z komnaty 2 mamy 3 możliwości z takim samym prawdopodobieństwem 1/3 że trafimy do kota, 1/3 - trafimy do 1 i 1/3 - trafimy do 5. a z komnaty 1 i 5 prawdopodobienstwo ze trafimy do kota to 1/2 dla każdej z nich. reasumując prawdopodobieństwo, że trafimy do kota z komnaty 2 jest równe 1/3+1/3*1/2+1/3*1/2=2/3.
Zbiorcze wyniki prezentujemy na diagramie poniżej:

```
GraphicsGrid[{{Style["kot: 1/2
ser: 1/2", 24], Style["kot: 2/3
ser: 1/3", 24], Style["kot: 1
ser: 0", 24]}, {Style["kot: 1/3
ser: 2/3", 24], Style["kot: 1/2
ser: 1/2", 24], Style["kot: 2/3
ser: 1/3", 24]}, {Style["kot: 0
ser: 1", 24], Style["kot: 1/3
ser: 2/3", 24], Style["kot: 1/2
ser: 1/2", 24]}}, Frame → All, ImageSize → 500]
```

kot: 1/2 ser: 1/2	kot: 2/3 ser: 1/3	kot: 1 ser: 0
kot: 1/3 ser: 2/3	kot: 1/2 ser: 1/2	kot: 2/3 ser: 1/3
kot: 0 ser: 1	kot: 1/3 ser: 2/3	kot: 1/2 ser: 1/2

Zad. V (symulacja szczura)

Losowanie początkowej komnaty (zakladamy, że szczur może mieć szzęscie i wylądować od razu z serem albo w kocie)

funkcja bladzenie[n] zwraca jedna z możliwych komnat do której może udać się szczur bedący w komnacie n

```
bladzenie[n_] := RandomChoice[Transpose[p][[n]] → Range[9]]
```

Funkcja symszczur[n] symuluje ruch szczura po labiryncie (szczur startuje w pokoju n).

```
symszcur[n_] := NestWhile[bladzenie[#] &, n, UnsameQ, 2]
```

sprawdzenie :

```
Table[symszcur[5], 10]
```

```
{7, 7, 7, 3, 7, 3, 3, 7, 3, 3}
```

Funkcja symulacja zwraca prawdopodobienstwo dojścia przez szczura do sera. Na próbę wpuszczamy do labiryntu k szcurów startujących z sali n.

```
symulacja[n_, k_] := Total[Table[If[symszcur[n] == 7, 1, 0], {i, 1, k}]] / k
```

Sprawdzamy prawdopodobienstwo dojscia do sera z punktu n przy ilosci sprawdzen k=10000

```
N[symulacja[9, 10000]]
N[symulacja[3, 10000]]
N[symulacja[7, 10000]]
```

```
0.4952

0.

1.
```

Wyniki zbiorcze dla różnych komnat : (k - ilosc szcurów)

```
zbiorcze[k_] := GraphicsGrid[
  {
   Table[Style[N[symulacja[i, k]], 32], {i, 1, 3}],
   Table[Style[N[symulacja[i, k]], 32], {i, 4, 6}],
   Table[Style[N[symulacja[i, k]], 32], {i, 7, 9}]
  }, Frame → All, ImageSize → 500]
```

Prawdopodobieństwo dojścia do sera z różnych komnat :

```
zbiorcze[2000]
```

0.5045	0.345	0.
0.6695	0.5015	0.3245
1.	0.6635	0.4935

Zad. VI (Dodatkowe)

Gęstość niezmiennicza dla rodziny logistycznej z parametrem $\lambda = 4$:

$$h(x) = \frac{1}{\pi\sqrt{x(1-x)}}$$

a) Dokładna wartość oczekiwana dana przez tą miarę :

$$N\left[\text{Integrate}\left[\text{Sqrt}[x] * \frac{1}{\pi * \text{Sqrt}\left[x * (1-x)\right]}, \{x, 0, 1\}\right]\right]$$

```
0.63662
```

b) Poprzez iteracje

Sprawdźmy, że miara absolutnie ciągła h (x) jest probabilistyczna

```
Integrate[1/(π * Sqrt[x * (1 - x)]), {x, 0, 1}]
```

```
1
```

Funkcja zwracająca orbitę o długości n dla rodziny logistycznej

```
orbitalogistyczna[x0_, n_] := NestList[N[4 * # * (1 - #)] &, x0, n - 1]
```

Losujemy punkt początkowy (dla p.w. elementow (0, 1) metoda powinna działać)

```
ppocz = RandomReal[];
```

Wyliczamy wartość $\frac{1}{n}\sum_{i=0}^{n-1} f\left(S^i(x)\right)$ dla dużych n, gdzie $f(x) = \sqrt{x}$.

```
aproksymacjalog[x0_, n_] := Total[Map[Sqrt, orbitalogistyczna[x0, n]]] / n
```

```
Table[q = aproksymacjalog[ppocz, 10^i]; "Dla n=" <>
  ToString[Superscript[10, ToString[i]], StandardForm] <> " wartość przybliżona to: " <>
  ToString[q] <> " błąd bezwzględny wynosi:" <> ToString[N[Abs[q - 2 / Pi]]], {i, 0, 7}]
```

```
{Dla n=10^0 wartość przybliżona to: 0.92115 błąd bezwzględny wynosi:0.28453,
 Dla n=10^1 wartość przybliżona to: 0.568822 błąd bezwzględny wynosi:0.0677976,
 Dla n=10^2 wartość przybliżona to: 0.641378 błąd bezwzględny wynosi:0.004758,
 Dla n=10^3 wartość przybliżona to: 0.640416 błąd bezwzględny wynosi:0.00379608,
 Dla n=10^4 wartość przybliżona to: 0.644671 błąd bezwzględny wynosi:0.00805074,
 Dla n=10^5 wartość przybliżona to: 0.633015 błąd bezwzględny wynosi:0.00360442,
 Dla n=10^6 wartość przybliżona to: 0.636765 błąd bezwzględny wynosi:0.000144812,
 Dla n=10^7 wartość przybliżona to: 0.636724 błąd bezwzględny wynosi:0.000104464}
```