

Mały Projekt 6 - Grafy

Jan Czechowski

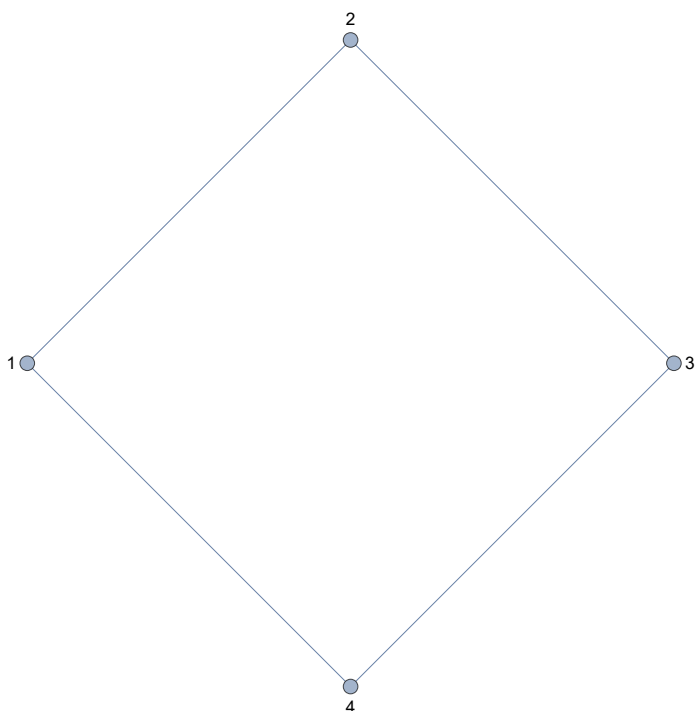
zadanie 1.

- Podaj przykład grafu, który jest jednocześnie hamiltonowski i eulerowski.

```
In[*]:= GraphPlot[CycleGraph[4], VertexLabels -> "Name"]
```

wykres grafu graf cykliczny etykiety wierzchołków

Out[*]=

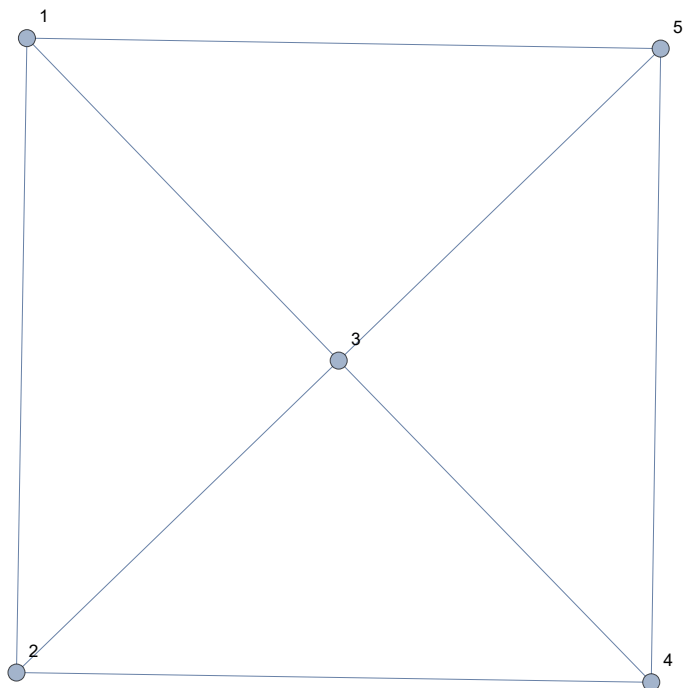


zadanie 2.

- Podaj przykład grafu, który jest hamiltonowski, ale nie jest eulerowski.

```
In[*]:= GraphPlot[Graph[{1 ↔ 2, 2 ↔ 3, 3 ↔ 4, 4 ↔ 5, 5 ↔ 1, 1 ↔ 3, 2 ↔ 4, 3 ↔ 5}],  
  wykres grafu |graf  
  VertexLabels → "Name"]  
  |etykiety wierzchołków
```

Out[*]=

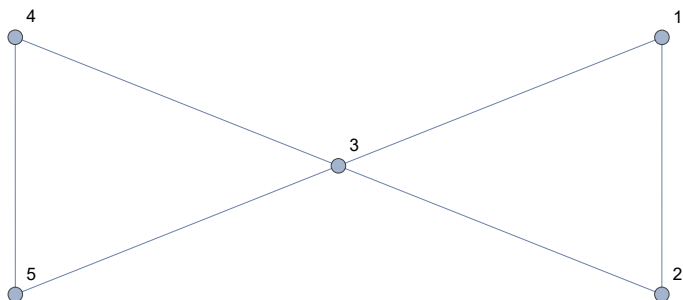


zadanie 3.

- Podaj przykład grafu, który jest eulerowski, ale nie jest hamiltonowski.

```
In[*]:= g = Graph[{1 ↔ 2, 2 ↔ 3, 3 ↔ 4, 4 ↔ 5, 5 ↔ 3, 3 ↔ 1}, VertexLabels → "Name"]  
  |graf |etykiety wierzchołków  
{FindEulerianCycle[g], FindHamiltonianCycle[g]}  
  |znajdź cykl Eulera |znajdź cykl Hamiltona
```

Out[*]=



Out[*]=

```
{{{1 ↔ 3, 3 ↔ 5, 5 ↔ 4, 4 ↔ 3, 3 ↔ 2, 2 ↔ 1}}, {}}
```

zadanie 4.

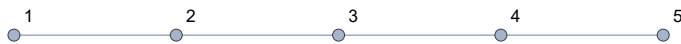
- Podaj przykład grafu, który nie jest ani hamiltonowski ani eulerowski.

```
In[*]:= GraphPlot[Graph[{1 ↔ 2, 2 ↔ 3, 3 ↔ 4, 4 ↔ 5}], VertexLabels → "Name"]
```

wykres grafu graf

etykiety wierzchołków

Out[*]=



zadanie 5.

- Znajdź kod Graya dla $n = 2, 3, 4, 5$.

```
In[*]:= GenerateGrayCode[1] := {{0}, {1}}
GenerateGrayCode[n_Integer] :=
    |liczba całkowita
Module[{prev, with0, with1}, prev = GenerateGrayCode[n - 1];
    |moduł
    with0 = Prepend[#, 0] & /@ prev;
    |dołącz do początku
    with1 = Prepend[#, 1] & /@ Reverse[prev];
    |dołącz do początku    |odwróć kolejność
    Join[with0, with1]
    |połącz

TableForm[Table[{"n = " <> ToString[n], Reverse[GenerateGrayCode[n]]}, {n, 2, 5}]]
```

w formie ta... tabela

przemień na ciąg... odwróć kolejność

```

Out[=]//TableForm=
      1 0
n = 2  1 1
      0 1
      0 0

      1 0 0
      1 0 1
n = 3  1 1 1
      1 1 0
      0 1 0
      0 1 1
      0 0 1
      0 0 0

      1 0 0 0
      1 0 0 1
      1 0 1 1
      1 0 1 0
      1 1 1 0
      1 1 1 1
      1 1 0 1
n = 4  1 1 0 0
      0 1 0 0
      0 1 0 1
      0 1 1 1
      0 1 1 0
      0 0 1 0
      0 0 1 1
      0 0 0 1
      0 0 0 0

      1 0 0 0 0
      1 0 0 0 1
      1 0 0 1 1
      1 0 0 1 0
      1 0 1 1 0
      1 0 1 1 1
      1 0 1 0 1
      1 0 1 0 0
      1 1 1 0 0
      1 1 1 0 1
      1 1 1 1 1
      1 1 1 1 0
      1 1 0 1 0
      1 1 0 1 1
      1 1 0 0 1
      1 1 0 0 0
n = 5  0 1 0 0 0
      0 1 0 0 1
      0 1 0 1 1
      0 1 0 1 0
      0 1 1 1 0
      0 1 1 1 1
      0 1 1 0 1
      0 1 1 0 0
      0 0 1 0 0
      0 0 1 0 1
      0 0 1 1 1
      0 0 1 1 0
      0 0 0 1 0
      0 0 0 1 1
      0 0 0 0 1
      0 0 0 0 0

```

zadanie 6.

- Wyznacz cykl Hamiltona dla grafu Graya o 8 oraz 16 wierzchołkach.

```

In[*]:= grafGraya[n_] := Graph[Range[0, 2^n - 1],
    Flatten[Table[UndirectedEdge[i, j], {i, 0, 2^n - 1}, {j, i + 1, 2^n - 1}], 1],
    VertexLabels -> "Name"]

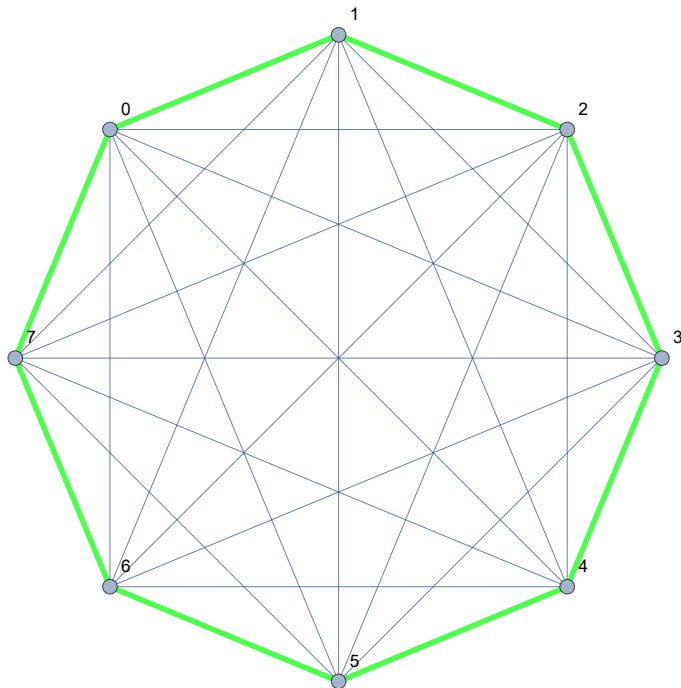
```

```

In[*]:= cyk18 = FindHamiltonianCycle[grafGraya[3]];
HighlightGraph[grafGraya[3], Style[cyk18, Thickness[0.008], Green]]

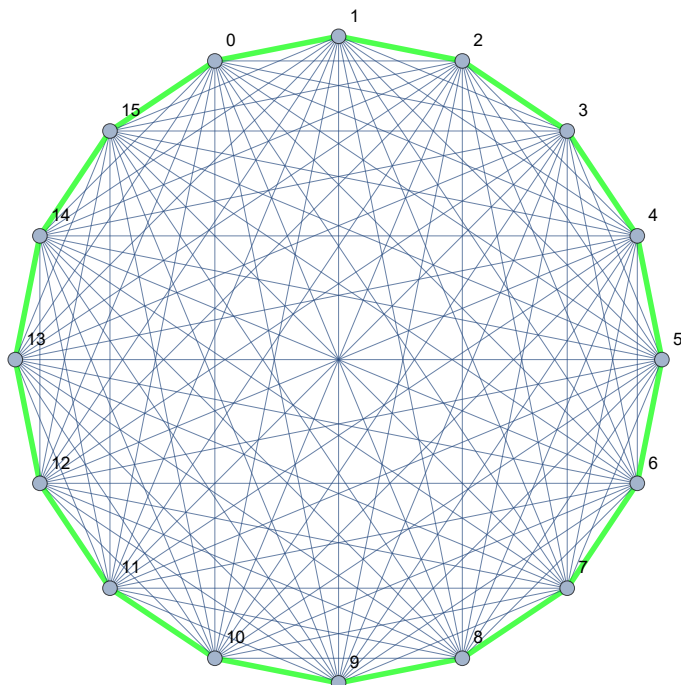
```

Out[*]=



```
In[*]:= cykl16 = FindHamiltonianCycle[grafGraya[4]];
           |znajdź cykl Hamiltona
HighlightGraph[grafGraya[4], Style[cykl16, Thickness[0.008], Green]]
           |uwypdatniony graf           |styl           |grubość           |zielony
```

Out[*]=



zadanie 7.

- Znajdź kod Prüfera dla wybranych drzew.

```
In[15]:= tree = TreeGraph[{1 -> 2, 1 -> 3, 2 -> 4, 2 -> 5}];
           |drzewo
In[16]:= pruferCode = ResourceFunction["LabeledTreeToPruferCode"][tree]
           |funkcja zasobów
```

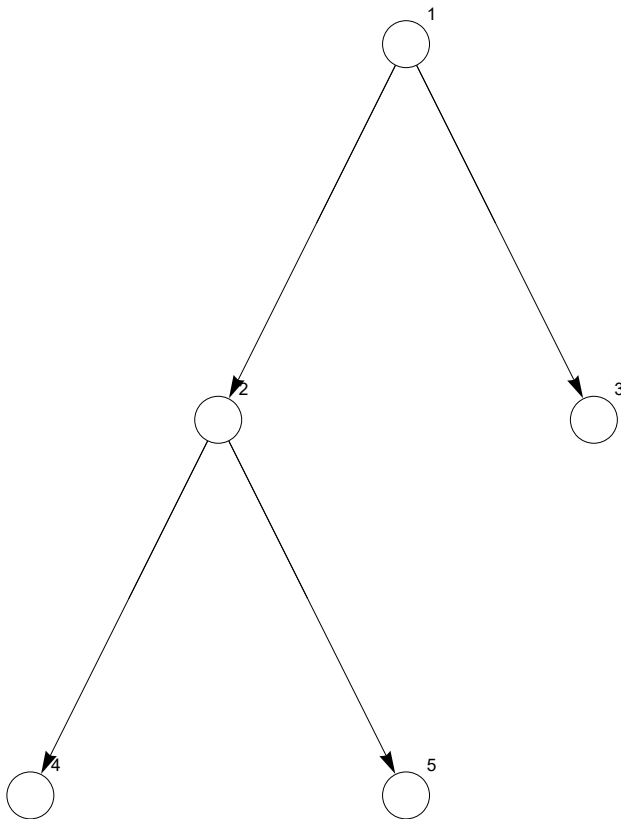
Out[16]=

{1, 2, 2}

```
In[17]:= TreeGraph[{1 → 2, 1 → 3, 2 → 4, 2 → 5}, VertexLabels → "Name", GraphStyle → "NameLabeled"]
```

[drzewo](#) [etykiety wierzchołków](#) [styl grafu](#)

Out[17]=



zadanie 8.

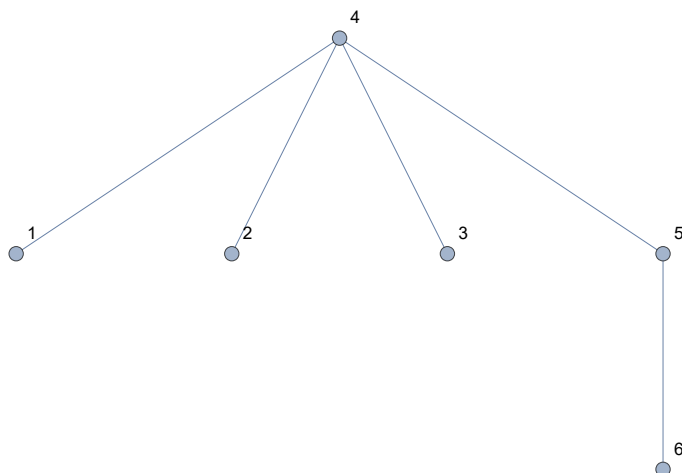
- Narysuj drzewo o podanym kodzie Prüfera.

```

In[*]:= TreeFromPrufer[pruferCode_List] :=
  Module[{n, vertices, degrees, edges = {}, pruferSequence = pruferCode},
    n = Length[pruferCode] + 2;
    vertices = Range[n];
    degrees = Table[1, {n}];
    Do[degrees[[v]]++, {v, pruferSequence}];
    While[Length[pruferSequence] > 0, v = First[Select[vertices, degrees[[#]] == 1 &]];
    AppendTo[edges, UndirectedEdge[v, First[pruferSequence]]];
    degrees[[v]]--;
    degrees[[First[pruferSequence]]]--;
    pruferSequence = Rest[pruferSequence];
    AppendTo[edges, UndirectedEdge@@Select[vertices, degrees[[#]] == 1 &]];
    Graph[vertices, edges, VertexLabels -> "Name"];

    pruferCode = {4, 4, 4, 5};
    tree = TreeFromPrufer[pruferCode]
  
```

Out[*]=



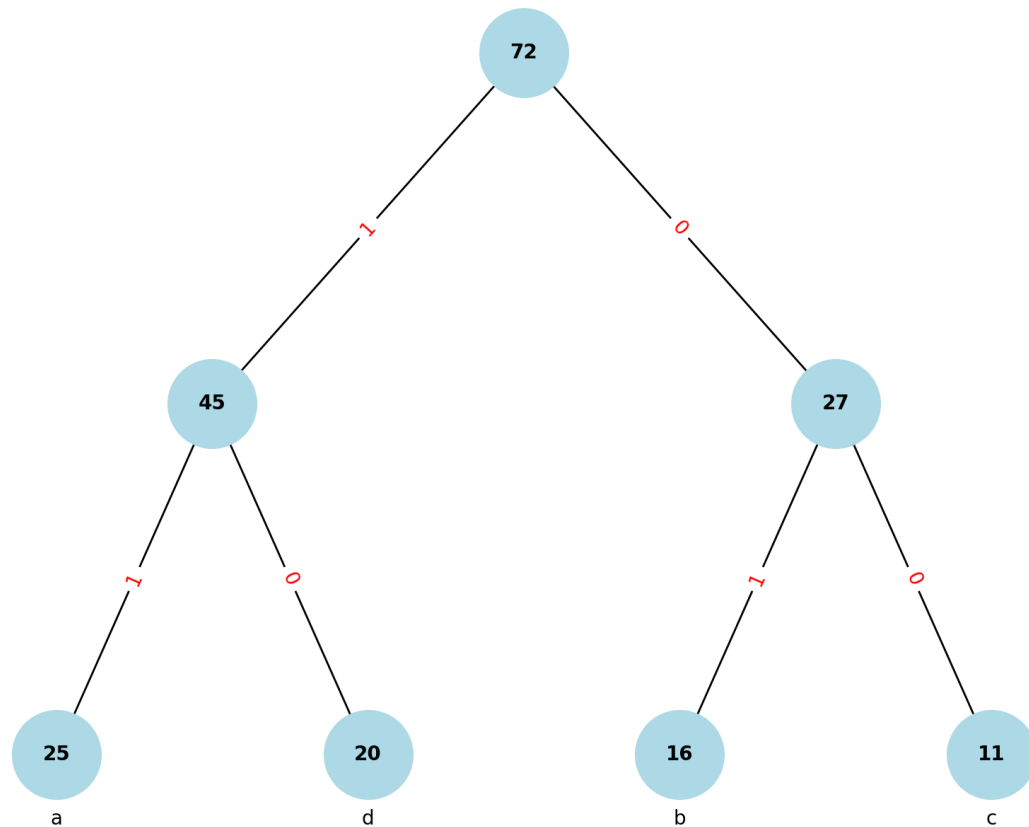
zadanie 9.

- Stosując kod Huffmana zakoduj wiadomość

$W = aaaaaabbbbabbcccccaaaacccbbaddaadddddadaaaaabbbdddddccccaaaaadddddddbbb.$


```
In[*]:= W = "aaaaabbbbabbccccccaaacbbaddaadddddaaaaabbbdddbcccaaaaaddddddbbbb";
czestotliwosci = Tally[Characters[W]]
           |zesta... |znaki w łańcuchu
```

```
Out[*]=
{{a, 25}, {b, 16}, {c, 11}, {d, 20}}
```



```
In[*]:= zamiana = <|"a" -> "11", "b" -> "01", "c" -> "00", "d" -> "10"|>;
```

```
zakodowanaWiadomosc = StringJoin[Riffle[zamiana[#, & /@ Characters[W], " "]]
           |połącz ciągi ... |przetasuj           |znaki w łańcuchu
```

```
Out[*]=
11 11 11 11 11 11 01 01 01 01 11 01 01 00 00 00 00 00 11 11 11 11 00
00 01 01 11 10 10 11 11 10 10 10 10 10 11 11 11 11 01 01 01 10 10 10
10 10 01 00 00 00 11 11 11 11 11 11 10 10 10 10 10 10 01 01 01 01
```

zadanie 10.

- Pewne miasteczko zamieszkują kobiety o tylko pięciu imionach: Anna, Barbara, Cecylia, Dorota i Elżbieta z częstotliwością odpowiednio 35%, 34%, 16%, 8% oraz 7%. Mamy ustalić imię losowo poznanej kobiety, zadając pytania, na które otrzymujemy odpowiedź jedynie **TAK** lub **NIE**. Jak należy pytać, aby średnia liczba pytań była najmniejsza?

Należy pytać tak jak według schematu poniżej aby średnia liczba pytań była najmniejsza. Ten schemat pozwoli ustalić imię losowo poznanej kobiety w 2 lub 3 pytaniach.

Pierwsze pytanie: "Czy to Anna lub Barbara?"

TAK→Przechodzimy do pytania 2A.

NIE→Przechodzimy do pytania 2B

Drugie pytanie (2A): "Czy to Anna?"

TAK→To Anna (35%).

NIE→To Barbara (34%).

Lub

Drugie pytanie (2B): "Czy to Cecylia?"

TAK→To Cecylia (16%).

NIE→Przechodzimy do pytania 3.

Trzecie pytanie: "Czy to Dorota?"

TAK→To Dorota (8%).

NIE→To Elżbieta (7%).