

(\*2. Считать данные из полученного входного файла в системе Wolfram Mathematica (учесть, что размерность графа может быть любой и параметры bi могут идти в произвольном порядке).\*)

```
In[1]:= inFileName = StringJoin[NotebookDirectory[], "input.txt"];
      |соединить с... |директория файла блокнота
fileStream = OpenRead[inFileName];
      |открыть для считывания
Is = Read[fileStream, {Word, Number}][[2]];
      |считать |слово |число
Us = Read[fileStream, {Word, Number}][[2]];
      |считать |слово |число
U = ReadList[fileStream, Expression, Us];
      |считать в список |выражение
weight = ReadList[fileStream, String, Is];
      |считать в список |строка
weight =
  Sort[Table[StringSplit[weight[[i]], {"b", "_", "/", "*"}, {"* / ", " "}], {i, Is}]];
      |сор... |табл... |разбить строку
UDirect = Table[U[[i, 1]] → U[[i, 2]], {i, Us}]
      |таблица значений
Close[fileStream];
      |закрыть
```

```
Out[8]= {1 ↔ 2, 3 ↔ 1, 1 ↔ 4, 5 ↔ 2, 1 ↔ 6, 3 ↔ 6, 3 ↔ 7, 4 ↔ 3, 5 ↔ 3, 5 ↔ 6, 6 ↔ 2, 7 ↔ 1, 7 ↔ 4}
```

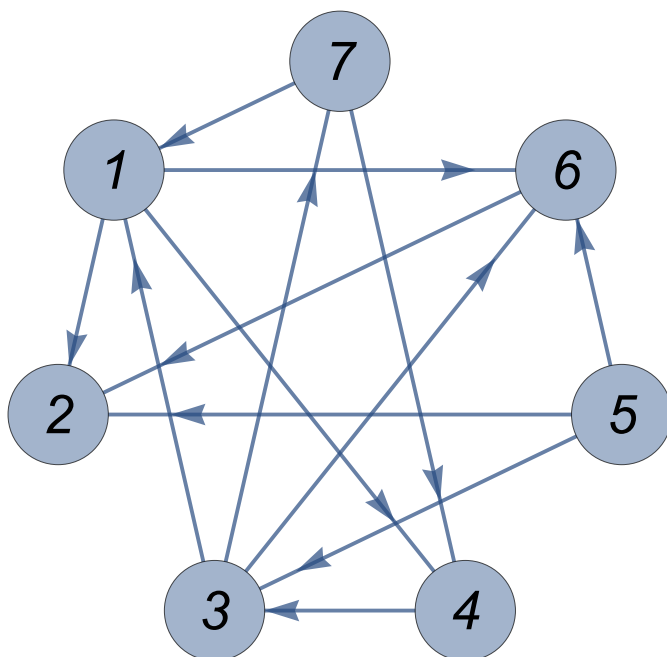
In[10]:=

```

(*3. По полученным данным создать ориентированный граф с заданием стилей/*)
g = Graph[UDirect, GraphLayout -> "CircularEmbedding",
  [граф [укладка графа]
  VertexLabels -> Placed["Name", Center], VertexSize -> 0.4,
  [метки для вершин [расположен [центр [размер вершины]
  VertexLabelStyle -> Directive[Italic, 28], EdgeShapeFunction ->
  [стиль меток вершин [директива [курсив [функция формы ребра]
  GraphElementData["FilledArrow", "ArrowSize" -> 0.05], EdgeStyle -> Thick]
  [стиль ребра [жирный]

```

Out[10]=



In[11]:=

```

(*4. Построить матрицу инцидентности для полученного графа.*)
(m = IncidenceMatrix[g]) // MatrixForm;
[матрица инцидентций [матричная форма]

```

# Лабораторная 3.

```
In[12]:= (*3. написать функцию построения системы уравнений баланса
для заданного графа с использованием функционального подхода/*)
edges = Subscript[x, #] & /@ UDirect;
      |с нижним индексом
equations = ConstantArray[0, Is];
      |постоянный массив
equations = Fold[With[{i = First[Last[#2]], j = Last[Last[#2]]},
      |све... |используя |первый|последний |пос... |последний
      ReplacePart[#1, {i → #1[[i]] + #2, j → #1[[j]] - #2}]] &, equations, edges];
      |заменить часть
equations // TableForm;
      |табличная форма
equationsSystem = equations[[#]] == ToExpression[weight[[#]][[2]]] & /@ Range[Is];
      |создать выражение |диапазон
equationsSystem // TableForm
      |табличная форма
```

Out[17]//TableForm=

```
x1→2 + x1→4 + x1→6 - x3→1 - x7→1 == 2
- x1→2 - x5→2 - x6→2 == -6
x3→1 + x3→6 + x3→7 - x4→3 - x5→3 == -4
- x1→4 + x4→3 - x7→4 == -1
x5→2 + x5→3 + x5→6 == 3
- x1→6 - x3→6 - x5→6 + x6→2 == 3
- x3→7 + x7→1 + x7→4 == 3
```

```
In[18]:= (*1. Используя процедурный подход,
написать функцию построения системы баланса в узлах графа.*)
equations = ConstantArray[0, Is];
      |постоянный массив
For[i = 1, i <= Us, i++, {t = UDirect[[i]],
      |цикл для
      equations[[t[[1]]]] += Subscript[x, t], equations[[t[[2]]]] -= Subscript[x, t]}}];
      |с нижним индексом |с нижним индексом
equationsSystem = ConstantArray[0, Is];
      |постоянный массив
For[i = 1, i <= Is, i++,
      |цикл для
      equationsSystem[[i]] = equations[[i]] == ToExpression[weight[[i]][[2]]]];
      |создать выражение
equationsSystem // TableForm
      |табличная форма
```

Out[22]//TableForm=

```
x1→2 + x1→4 + x1→6 - x3→1 - x7→1 == 2
- x1→2 - x5→2 - x6→2 == -6
x3→1 + x3→6 + x3→7 - x4→3 - x5→3 == -4
- x1→4 + x4→3 - x7→4 == -1
x5→2 + x5→3 + x5→6 == 3
- x1→6 - x3→6 - x5→6 + x6→2 == 3
- x3→7 + x7→1 + x7→4 == 3
```

```
In[30]:= (*2. Решить полученную систему функцией Solve
          |решить уравнения
и проверить правильность решения функцией Simplify
          |упростить
(функцией упрощения системы, после подстановки в нее полученного решения).*)
s = Solve[equationsSystem]
      |решить уравнения
Simplify[equationsSystem /. s[[1]]]
      |упростить

Out[30]= { {X4→3 → 4 - X1→2 - X1→6 + X3→1 + X3→7, X5→3 → X1→2 + X1→6 + X3→6,
           X5→6 → 3 - X1→2 - X1→6 - X3→6 - X5→2, X6→2 → 6 - X1→2 - X5→2,
           X7→1 → -2 + X1→2 + X1→4 + X1→6 - X3→1, X7→4 → 5 - X1→2 - X1→4 - X1→6 + X3→1 + X3→7 } }

Out[31]= {True, True, True, True, True, True, True}
```