

```

In[65]:= inFileName = StringJoin[{NotebookDirectory[], "input.txt"}];
           |соединить строки| |директория файла блокнота|
fileStream = OpenRead[inFileName];
           |открыть для считывания|
vertex = Read[fileStream, {Word, Number}][[2]];
           |считать| |слово| |число|
edge = Read[fileStream, {Word, Number}][[2]];
           |считать| |слово| |число|
edges = ReadList[fileStream, Expression, edge];
           |считать в список| |выражение|
vertexList = Array[# &, vertex];
           |массив|
edgesList = Table[edges[[i, 1]] → edges[[i, 2]], {i, edge}];
           |таблица значений|
graph = Graph[vertexList, edgesList, {GraphLayout → "CircularEmbedding",
           |граф| |укладка графа|
           VertexSize → 0.3, VertexLabels → Placed["Name", Center],
           |размер вершины| |метки для вершин| |расположен| |центр|
           VertexLabelStyle → Directive[Bold, Italic, 20],
           |стиль меток вершин| |директива| |жир| |курсив|
           EdgeShapeFunction → GraphElementData["Arrow", "ArrowSize" → 0.05]}}];
           |функция формы ребра| |стрелка|

pred = ConstantArray[0, vertex];
           |постоянный массив|
depth = ConstantArray[0, vertex];
           |постоянный массив|
dir = ConstantArray[0, vertex];
           |постоянный массив|
listUt = {};
dinast = {};

dinastVertex = {};
prev = root;
BuildSpanningTreeForGraph[g_, root_] := Module[{s = {}},
           |программный модуль|
           DepthFirstScan[UndirectedGraph[g], root,
           |проход в глубину| |ненаправленный граф|
           {"FrontierEdge" → Function[e, {AppendTo[s, e[[1]] → e[[2]]],
           |функция| |добавить в конец к|
           pred[[e[[2]]]] = e[[1]], depth[[e[[2]]]] = 1 + depth[[e[[1]]]]}],
           "PrevisitVertex" → Function[u, AppendTo[dinastVertex, u]]}];
           |функция| |добавить в конец к|
           For[k = 1, k ≤ Length[s], k++, arc = s[[k]];
           |цикл для| |длина|
           If[MemberQ[edgesList, arc], {dir[[arc[[2]]]] = 1;
           |...| |элемент списка?|
           AppendTo[listUt, arc]}, {dir[[arc[[2]]]] = -1;
           |добавить в конец к|
           AppendTo[listUt, Reverse[arc]]}]];
           |добавить в конец к| |расположить в обратном порядке|

```

```

Return[s];];
|вернуть управление

root = 5;
(*1,3*)
DFS = BuildSpanningTreeForGraph[graph, root];
(*2*)
Print["Множество дуг покрывающего дерева"]
|печатать
listUt
Print["Множество дуг, которые не вошли в покрывающее дерево"]
|печатать
listUn = Complement[edgesList, listUt] (*Un=U\Ut*)
|дополнение
(*4*)
graph
HighlightGraph[graph, listUt, VertexLabels -> "Some"]
|граф с подкраской |метки для вершин
HighlightGraph[graph, listUn, VertexLabels -> "Some"]
|граф с подкраской |метки для вершин
Print["Покрывающее дерево"]
|печатать
Graph[listUt, GraphLayout -> {"LayeredDigraphEmbedding", "RootVertex" -> root},
|граф |укладка графа
VertexSize -> 0.5, VertexLabels -> Placed["Name", Center],
|размер вершины |метки для вершин |расположен |центр
VertexLabelStyle -> Directive[Bold, Italic, 20],
|стиль меток вершин |директива |жи... |курсив
EdgeShapeFunction -> GraphElementData["Arrow", "ArrowSize" -> 0.12]]
|функция формы ребра |стрелка
Print["Корневое дерево с подсвеченным корнем"]
|печатать
TreeGraph[DFS, GraphLayout -> {"LayeredDigraphEmbedding", "RootVertex" -> root},
|граф дерево |укладка графа
VertexSize -> 0.5, VertexLabels -> Placed["Name", Center],
|размер вершины |метки для вершин |расположен |центр
VertexStyle -> {root -> Red}, VertexLabelStyle -> Directive[Bold, Italic, 20],
|стиль вершины |крас... |стиль меток вершин |директива |жи... |курсив
EdgeShapeFunction -> GraphElementData["Arrow", "ArrowSize" -> 0.12]]
|функция формы ребра |стрелка
all = {Prepend[vertexList, "vertex"], Prepend[pred, "pred"],
|добавить в начало |добавить в начало
Prepend[dir, "dir"], Prepend[depth, "depth"], Prepend[dinastVertex, "d"]};
|добавить в начало |добавить в начало |добавить в начало
Print["Root = ", root]
|печатать |корень уравнения
Text[Grid[all, Alignment -> Left, Spacings -> {2, 1}, Frame -> All]]
|текст |таблица |выравнивание |слева |размер зазора |рамка |всё
Множество дуг покрывающего дерева

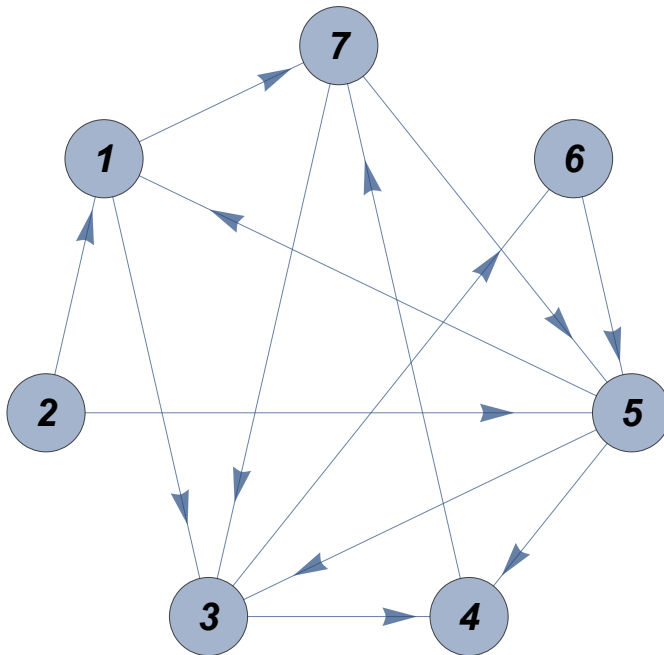
```

Out[84]= { 2 \leftrightarrow 5, 2 \leftrightarrow 1, 1 \leftrightarrow 3, 7 \leftrightarrow 3, 4 \leftrightarrow 7, 3 \leftrightarrow 6 }

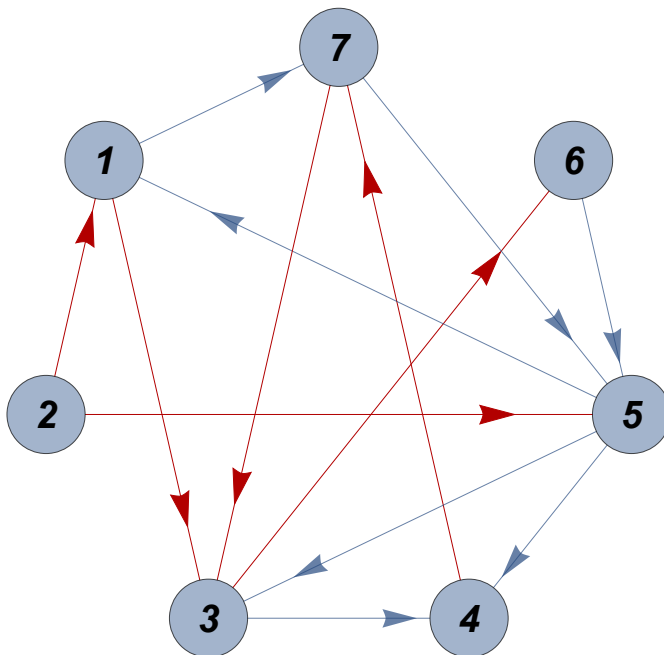
Множество дуг, которые не вошли в покрывающее дерево

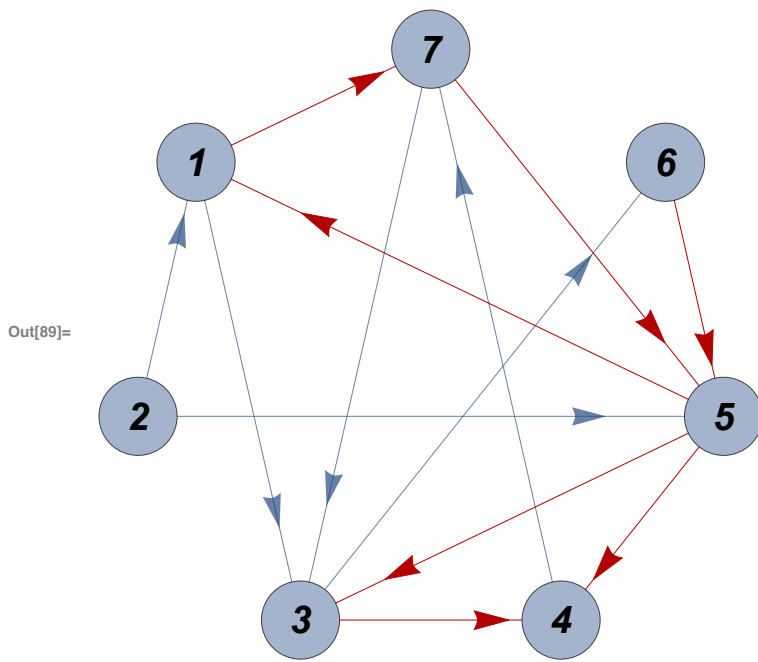
Out[86]= { 1 \leftrightarrow 7, 3 \leftrightarrow 4, 5 \leftrightarrow 1, 5 \leftrightarrow 3, 5 \leftrightarrow 4, 6 \leftrightarrow 5, 7 \leftrightarrow 5 }

Out[87]=

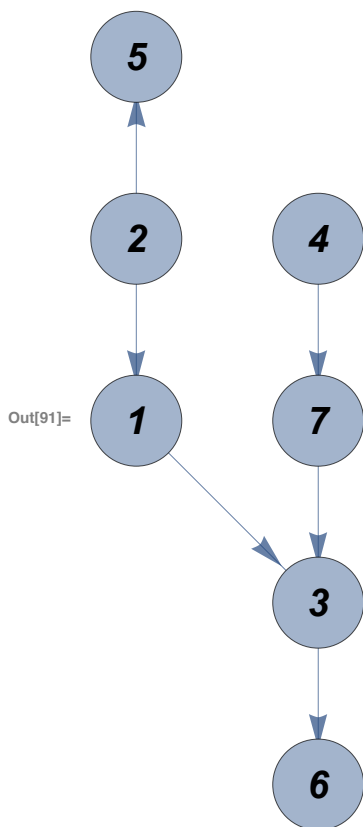


Out[88]=



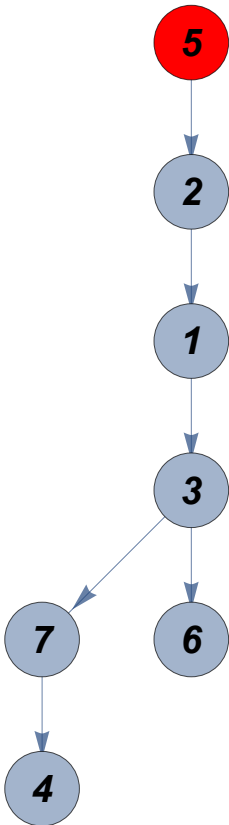


Покрывающее дерево



Корневое дерево с подсвеченным корнем

Out[93]=



Root = 5

Out[96]=

vertex	1	2	3	4	5	6	7
pred	2	5	1	7	0	3	3
dir	1	-1	1	-1	0	1	-1
depth	2	1	3	5	0	4	4
d	5	2	1	3	7	4	6

In[]:=

ln[\oplus]:=