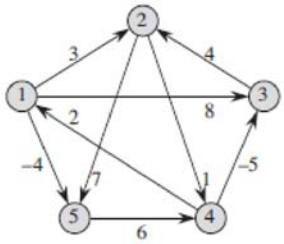
Trumpiausio kelio radimas

A.Domarkas, VU

Čia naudojama atvirojo kodo kompiuterinės algebros programa Maxima 5.31.2

1 pavyzdys([1], Figure 25.1) Rasime trumpiausius kelius tarp visų viršūnių.

Figure 1:




```
(%i44) load(graphs)$
```

```
(%i2) g:create_graph([1,2,3,4,5],
      [[[1,2],3],[[1,3],8],[[1,5],-4],
       [[2,4],1],[[2,5],7],
       [[3,2],4],
       [[4,1],2],[[4,3],-5],
       [[5,4],6]],
      'directed=true
      );
(%o2) DIGRAPH(5 vertices, 9 arcs)
```

Pavyzdžiui, rasime trumpiausią kelią nuo 3 viršūnės iki 1 viršūnės:

```
(%i3) shortest_weighted_path(3,1,g);
(%o3) [7,[3,2,4,1]]
```

Todėl trumpiausio kelio nuo 3 viršūnės iki 1 viršūnės svoris yra 7, o maršrutas yra 3 -> 2 -> 4 -> 1.

```
(%i4) kill(h)$
```

```
(%i5) h[i,j]:=shortest_weighted_path(i,j,g)[1]$
```

```
(%i6) genmatrix(h,5,5);
```

(%o6) $\begin{bmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix}$

Gavome tą pačią matricą D(5), kaip ir [1], brėž. 25.4
Jei norime matyti maršrutus, tai galima taip:

```
(%i7) h1[i,j]:=shortest_weighted_path(i,j,g)$
```

```
(%i8) genmatrix(h1,5,5);
```

(%o8) $\begin{bmatrix} [0,[1]] & [1,[1,5,4,3,2]] & [-3,[1,5,4,3]] & [2,[1,5,4]] & [-4,[1,5]] \\ [3,[2,4,1]] & [0,[2]] & [-4,[2,4,3]] & [1,[2,4]] & [-1,[2,4,1,5]] \\ [7,[3,2,4,1]] & [4,[3,2]] & [0,[3]] & [5,[3,2,4]] & [3,[3,2,4,1,5]] \\ [2,[4,1]] & [-1,[4,3,2]] & [-5,[4,3]] & [0,[4]] & [-2,[4,1,5]] \\ [8,[5,4,1]] & [5,[5,4,3,2]] & [1,[5,4,3]] & [6,[5,4]] & [0,[5]] \end{bmatrix}$

2 būdas. Sprendimas Floid-Warshall algoritmu ([1], 25.2 skyrelis)

```
(%i9) W:matrix(
[0,3,8,inf,-4],
[inf,0,inf,1,7],
[inf,4,0,inf,inf],
[2,inf,-5,0,inf],
[inf,inf,inf,6,0]
);
```

(%o9) $\begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$

Pagal [1], 25.5 formulę sudarome komandą

```
(%i10) D(k):=block([h,m,n,d],kill(h),
[m,n]:matrix_size(W),
d(i,j,k):=if k=0 then W[i,j] else
min(d(i,j,k-1),d(i,k,k-1)+d(k,j,k-1)),
h[i,j]:=d(i,j,k),
genmatrix(h,m,n))$
```

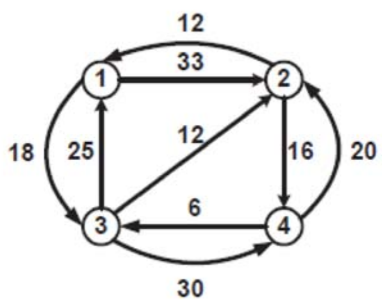
```
(%i11) makelist(D(k),k,0,5);
```

(%o11) $\left[\begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix} \right]$

Gavome tą pačią matricų seką D(k), kaip ir [1], brėž. 25.4

2 pavyzdys ([2], p. 205, 5.4 pavyzdys arba [4])
Rasime trumpiausius kelius tarp visų grafo viršūnių.

Figure 2:



```
(%i12) W:matrix([0,33,18,inf],[12,0,inf,16],[25,12,0,30],[inf,20,6,0]);
```

(%o12) $\begin{bmatrix} 0 & 33 & 18 & \infty \\ 12 & 0 & \infty & 16 \\ 25 & 12 & 0 & 30 \\ \infty & 20 & 6 & 0 \end{bmatrix}$

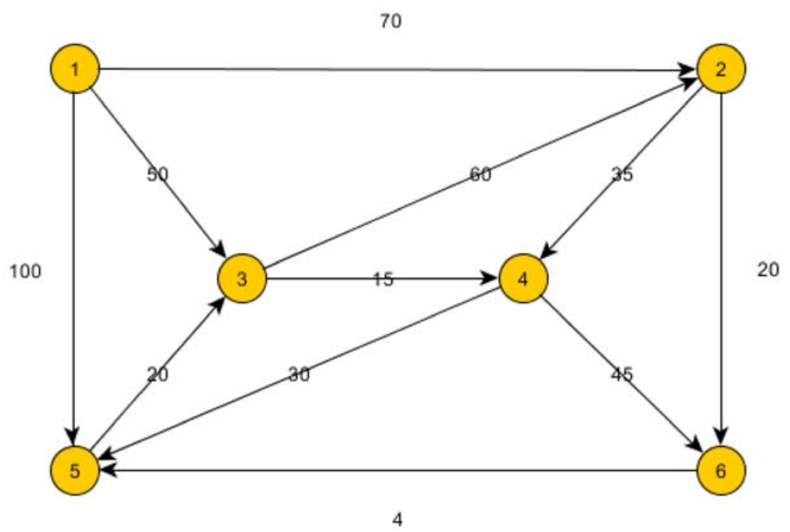
```
(%i13) makelist(D(k),k,0,4);
```

(%o13) $\left[\begin{bmatrix} 0 & 33 & 18 & \infty \\ 12 & 0 & \infty & 16 \\ 25 & 12 & 0 & 30 \\ \infty & 20 & 6 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 33 & 18 & \infty \\ 12 & 0 & 30 & 16 \\ 25 & 12 & 0 & 30 \\ \infty & 20 & 6 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 33 & 18 & 49 \\ 12 & 0 & 30 & 16 \\ 24 & 12 & 0 & 28 \\ 32 & 20 & 6 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 30 & 18 & 46 \\ 12 & 0 & 30 & 16 \\ 24 & 12 & 0 & 28 \\ 30 & 18 & 6 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 30 & 18 & 46 \\ 12 & 0 & 22 & 16 \\ 24 & 12 & 0 & 28 \\ 30 & 18 & 6 & 0 \end{bmatrix} \right]$

Gavome tą pačią matricų seką D(k), kaip ir [2], p. 205.
Palyginkite šią sprendimo programą su [2], p. 204.
Kuri yra trumpesnė ir lengviau realizuojama ?
Dirbant su didesniais grafais programa iš [2] gali būti greitesnė.

3 pavyzdys ([3], p. 197 arba [4]) Rasime trumpiausius kelius iš 1 viršūnės iki likusių grafo viršūnių.

Figure 3:




```
(%i14) load(graphs)$
```

```
(%i15) gr:create_graph([1,2,3,4,5,6],
       [[[1,2],70],[[1,3],50],[[1,5],100],
       [[2,4],35],[[2,6],20],
       [[3,2],60],[[3,4],15],
       [[4,5],30],[[4,6],45],[[6,5],4]],
       'directed=true)$
```

```
(%i16) c2:shortest_weighted_path(1, 2, gr);
(%o16) [70,[1,2]]
```

```
(%i17) c3:shortest_weighted_path(1, 3, gr);
(%o17) [50,[1,3]]
```

```
(%i18) c4:shortest_weighted_path(1, 4, gr);
(%o18) [65,[1,3,4]]
```

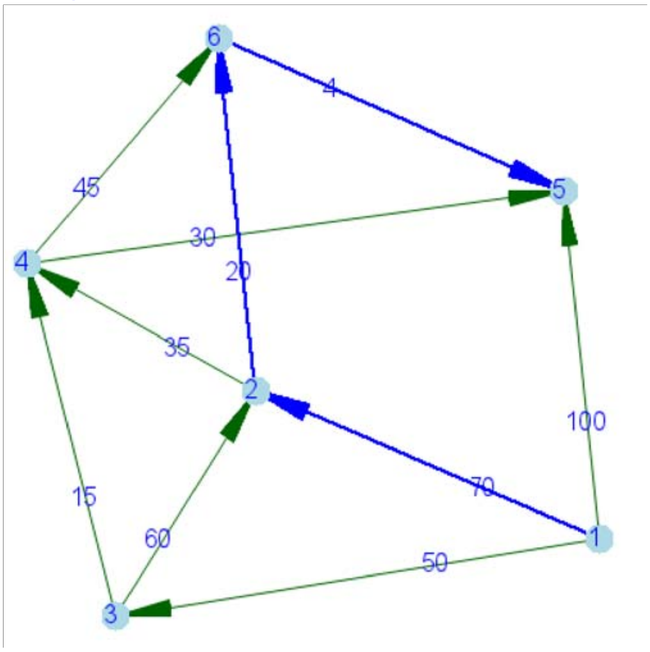
```
(%i19) c5:shortest_weighted_path(1, 5, gr);
(%o19) [94,[1,2,6,5]]
```

```
(%i20) c6:shortest_weighted_path(1, 6, gr);
(%o20) [90,[1,2,6]]
```

Nubrėžiame grafą ir kelią nuo 1 iki 5 (c5):

```
(%i21) draw_graph(
           gr,
           show_weight=true,
           show_id=true,
           show_vertices=[1,2,3,4,5,6],
           head_length=0.1,
           head_angle=8,
           edge_color="dark-green",
       text_color=blue,
       show_edge_width=2,
       show_edges=vertices_to_path(c5[2])
           )$
```

(%t21)



4 pavyzdys. (iš Grimaldi vadovėlio [2] p. 658-662)
Rasti trumpiausius kelius nuo viršūnės c iki kitų viršūnių a, b, c, f, g, h .
Grafas vadovėlyje pavaizduotas taip:

Figure 4:

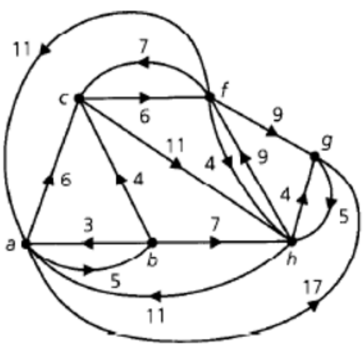


Figure 13.1

```
(%i22) load(graphs)$
```

```
(%i23) gr:create_graph([1,2,3,4,5,6],
       [[[1,2],5],[[1,3],6],[[1,5],17],
       [[2,3],4],[[2,6],7], [[2,1],3],
       [[3,4],6],[[4,3],7],[[3,6],11],
       [[4,1],11],[[4,5],9],[[4,6],4],
       [[5,6],5],[[6,4],9],[[6,5],4],[[6,1],11]],
       'directed=true)$
```

```
(%i24) set_vertex_label(1, "a", gr)$
       set_vertex_label(2, "b", gr)$
       set_vertex_label(3, "c", gr)$
       set_vertex_label(4, "f", gr)$
       set_vertex_label(5, "g", gr)$
       set_vertex_label(6, "h", gr)$
```

Randame trumpiausius kelius nuo c(3 viršūnė) iki kitų viršūnių a, b, c, f, g, h (1, 2, 3, 4, 5, 6):

```
(%i30) c1:shortest_weighted_path(3, 1, gr);
(%o30) [17,[3,4,1]]
```

```
(%i31) c2:shortest_weighted_path(3, 2, gr);
(%o31) [22,[3,4,1,2]]
```

```
(%i32) c3:shortest_weighted_path(3, 3, gr);
(%o32) [0,[3]]
```

```
(%i33) c4:shortest_weighted_path(3, 4, gr);
(%o33) [6,[3,4]]
```

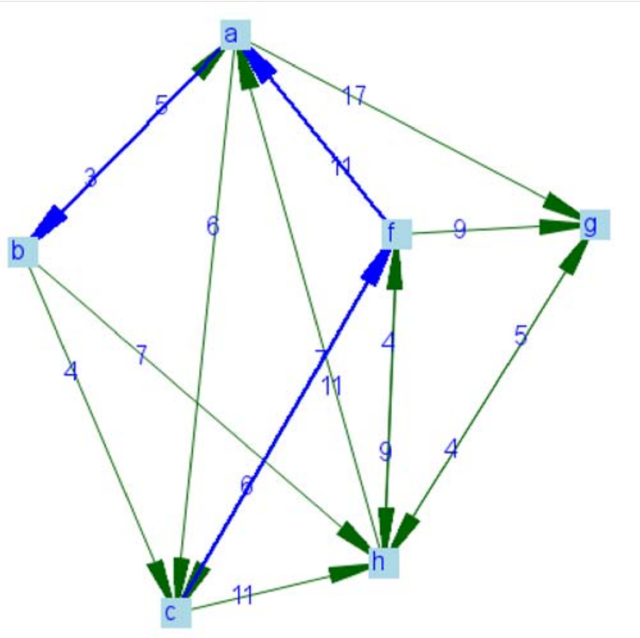
```
(%i34) c5:shortest_weighted_path(3, 5, gr);
(%o34) [14,[3,4,6,5]]
```

```
(%i35) c6:shortest_weighted_path(3, 6, gr);
(%o35) [10,[3,4,6]]
```

Nubrėžiame grafą ir kelią nuo c iki b (c2):

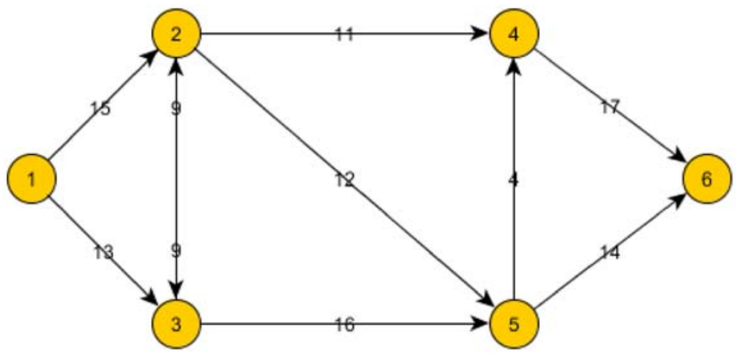
```
(%i36) draw_graph(
           gr,
           show_weight=true,
       show_label=true,
           vertex_size=0,
           show_vertices=[1,2,3,4,5,6],
           show_vertex_type=filled_square,
           head_length=0.1,
           head_angle=8,
           edge_color="dark-green",
       text_color=blue,
       show_edge_width=2,
       show_edges=vertices_to_path(c2[2])
           )$
```

(%t36)



5 pavyzdys. ([5]), example 6.14) Rasime trumpiausius kelią nuo 1-os iki 6-os viršūnės.

Figure 5:



1 būdas

```
(%i37) load(simplex)$
```

Nežinomuosius pažymėkime xij,
xij = 1, jei vykstame keliu iš i-osios viršūnės į j-ąją viršūnę,
xij = 0, jei ne.

Reikia minimizuoti funkciją

```
(%i38) f:15*x12+13*x13+9*x23+9*x32+11*x24+12*x25+16*x35+4*x54+17*x46+14*x56;
```
(%o38) $14\,x56+4\,x54+17\,x46+16\,x35+9\,x32+12\,x25+11\,x24+9\,x23+13\,x13+15\,x12$

kai(žr. [5])

```
(%i39) apr:[x12+x32=x24+x25+x23,x13+x23=x32+x35,x24+x54=x46,x35+x25=x54+x56,x12+x23=1,x46+x56=1];
```
(%o39) $[x32+x12=x25+x24+x23 , x23+x13=x35+x32 , x54+x24=x46 , x35+x25=x56+x54 , x23+x12=1 , x56+x46=1]$

```
(%i40) minimize_lp(f,apr),nonegative_lp=true;
```
(%o40) $[41 , [x56=1 , x46=0 , x54=0 , x35=0 , x13=0 , x25=1 , x24=0 , x23=0 , x32=0 , x12=1]]$

Atsakymas:
Trumpiausias maršrutas yra 1 -> 2 -> 5 -> 6 ; jo ilgis 41 kilometras.

2 būdas

```
(%i41) load(graphs)$
```

```
(%i42) gr:create_graph([1,2,3,4,5,6],
       [[[1,2],15],[[1,3],13],
       [[2,3],9],[[2,4],11],[[2,5],12],
       [[3,2],9], [[3,5],16],
       [[4,6],17],
       [[5,4],4], [[5,6],14]],
       'directed=true)$
```

Randame trumpiausius kelius iš 1-os iki 6-os viršūnės:

```
(%i43) shortest_weighted_path(1, 6, gr);
```
(%o43) $[41 , [1 , 2 , 5 , 6]]$

Todėl trumpiausio kelio ilgis yra 41 km, o maršrutas yra 1 -> 2 -> 5 -> 6.

Literatūra:

[1] T.H.Cormen e. a., Introduction to algorithms, 3rd ed., 2009, ch. 23
[2] R. P. Grimaldi, Discrete and Combinatorial Mathematics: An Applied Introduction, third edition, 1994
[3] R.Čiegis, Duomenų struktūros, algoritmai ir jų analizė, Vilnius, Technika, 2007
[4] R.Čiegis http://www.techmat.vgtu.lt/konspektai/Algoritmai/Paskaita9.pdf
[5] M. Asghar Bhatti. Practical Optimizations Methods with Mathematica Applications, Springer, New York, 2000