# GraphTheory

A substantial effort was put into Graph Theory for Maple 2023, including improved ability to solve traveling salesman problems, support for multigraphs, new commands for graph computation, and advances in visualization.

> with(GraphTheory):

**Traveling Salesman** 

Multigraph support

**Graph products** 

Additions to SpecialGraphs

### **Traveling Salesman**

The <u>TravelingSalesman</u> command now makes use of <u>Concorde</u>, a well-known library implementing highly efficient heuristics for solving instances of the traveling salesman problem. This addition considerably increases the size of problems which TravelingSalesman is able to handle.

#### Example: Using TravelingSalesman with vertexpositions

In the following example, we import a dataset of the 100 largest cities in the continent of Africa (including the island of Madagascar) and ask TravelingSalesman to find a minimal tour of them.

> AfricanCities := Import("example/AfricanCities100.tsv", base = datadir, output = Matrix, skiplines = 1)

```
"Cairo"
                                                  "Egypt"
                                                                       22183200
                                                                                  30.044444
                                                                                               31.235833
                  2
                       "Kinshasa"
                                     "Democratic Republic of the Congo" 16315534
                                                                                   -4.325
                                                                                              15.322222
                  3
                        "Lagos"
                                                 "Nigeria"
                                                                       15387639
                                                                                  6.455027
                                                                                               3.384082
                  4
                         "Giza"
                                                 "Egypt"
                                                                       9200000
                                                                                   29.987
                                                                                               31.2118
                                                                                 -8.838333 13.234444
                        "Luanda"
                                                 "Angola"
                                                                       8952496
                     "Dar es Salaam"
                                                "Tanzania"
                                                                       7404689
                                                                                 -6.816111 39.280278
AfricanCities :=
                       "Khartoum"
                                                 "Sudan"
                                                                                  15.500556
                                                                                                32.56
                                                                       6160327
                     "Johannesburg"
                                               "South Africa"
                                                                       6065354 - 26.204444 28.045556
                       "Abidjan"
                  9
                                              "Côte d'Ivoire"
                                                                       5515790
                                                                                  5.316667
                                                                                             -4.033333
                      "Alexandria"
                                                  "Egypt"
                                                                       5483605
                                                                                   31.1975
                                                                                               29.8925
                                                                                                 100 × 6 Matrix
```

```
G := CompleteGraph(convert(AfricanCities[..., 2], list), vertexpositions = AfricanCities[..., [6, 5]])
```

 $G := Graph \ 1$ : an undirected graph with 100 vertices and 4950 edge(s)

With the new vertexpositions option, TravelingSalesman uses edge weights computed from the geometric distance between vertices in the graph layout specified previously when CompleteGraph was called. The new startvertex option allows a particular starting vertex to be specified.

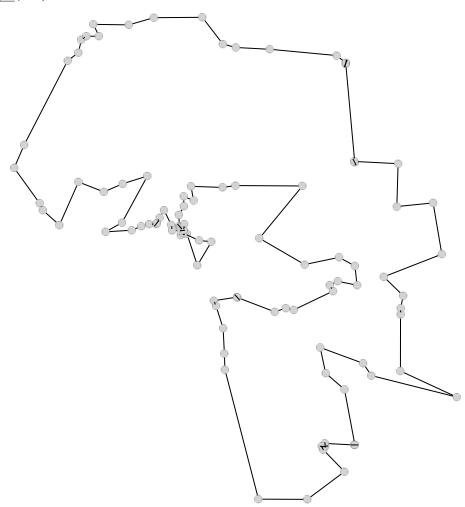
```
> W, T := TravelingSalesman (G, vertexpositions, startvertex = "Cairo")

W, T := 421.032920193835, ["Cairo", "Alexandria", "Benghazi", "Misratah", "Tripoli", "Tunis",
    "Algiers", "Oran", "Tangier", "Fez", "Rabat", "Casablanca", "Marrakesh", "Agadir", "Nouakchott",
    "Dakar", "Conakry", "Freetown", "Monrovia", "Bamako", "Bobo Dioulasso", "Ouagadougou",
    "Niamey", "Ilorin", "Ibadan", "Kumasi", "Abidjan", "Accra", "Lomé", "Abomey-Calavi", "Lagos",
    "Ikorodu", "Benin City", "Warri", "Owerri", "Umuahia", "Onitsha", "Nnewi", "Port Harcourt",
    "Yaoundé", "Douala", "Libreville", "Uyo", "Aba", "Enugu", "Lokoja", "Abuja", "Kaduna", "Jos",
    "Kano", "Maiduguri", "N'Djamena", "Nyala", "Bangui", "Kisangani", "Bunia", "Kampala",
    "Mwanza", "Kigali", "Bujumbura", "Bukavu", "Mbuji-Mayi", "Kananga", "Tshikapa", "Kinshasa",
    "Brazzaville", "Pointe-Noire", "Cabinda", "Luanda", "Benguela", "Lubango", "Cape Town",
    "Gqeberha (Port Elizabeth)", "Durban (eThekwini)", "Vereeniging (Emfuleni)", "West Rand",
    "Johannesburg", "East Rand (Ekurhuleni)", "Pretoria (Tshwane)", "Matola", "Maputo", "Harare",
    "Lusaka", "Lubumbashi", "Lilongwe", "Blantyre", "Antananarivo", "Nampula", "Dar es Salaam",
    "Zanzibar", "Mombasa", "Nairobi", "Mogadishu", "Hargeisa", "Addis Ababa", "Asmara",
    "Omdurman", "Khartoum", "Giza", "Shubra el-Kheima", "Cairo"]
```

We will illustrate the tour by constructing a subgraph consisting only of the edges included in the optimal tour across G.

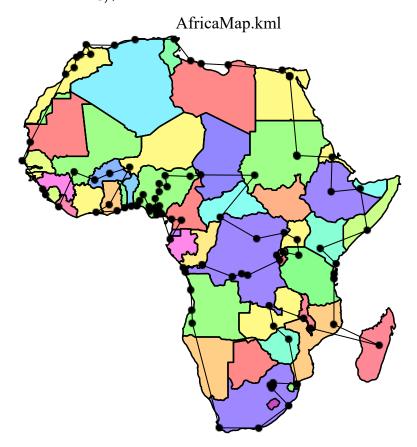
>  $TG := Subgraph(\ G, Trail(T)\ );$   $TG := Graph\ 2: \ an\ undirected\ graph\ with\ 100\ vertices\ and\ 100\ edge(s)$ 

> <u>DrawGraph</u>(TG)



To better visualize the tour we can combine this with a country map of Africa.

> plots:-display(Import("example/AfricaMap.kml", base = datadir), DrawGraph(TG, stylesheet = [vertexcolor = black]))



#### Using TravelingSalesman with an arbitrary weight matrix

The previous example demonstrates the use of TravelingSalesman with edge weights corresponding directly to geometric distances between vertices. In many instances of the traveling salesman problem, the weights do not correspond so directly to the geometry.

Fortunately, TravelingSalesman can also compute a tour when given an arbitrary weight matrix. To illustrate this, let us extend the previous example of a tour of 100 African cities to incorporate the idea that there might be some increased cost associated with traversing an international border.

First let us get the matrix of Euclidean distances from the previous example:

> DM := WeightMatrix(MakeWeighted(G, vertexpositions, metric = Euclidean))

```
DM :=
                                                                                                   37.8748161174210
                                                                                                                                                                   16.0850293372542 37.8125909323612 4.97282532833931 24.0872182145107 26.2715370902888 25.3099446428611 21.6240433753013 38.3945439775925
                            36.4990496885315 16.0850293372542
                                                                                                                                                                                                                                               36.4435953544687
                                                                                                                                                                                                                                                                                                                      18.1911102360643 38.2708765392885 30.5459618613093 40.9246789340187 7.50425937530314 36.2613595307519
                        0.0622687580171630 37.8125909323612 36.4435953544687
                                                                                                                                                                                                                                                                                                                        42.8476411621579 4.97282532833931 18.1911102360643 42.7854158716685
                                                                                                                                                                                                                                                                                                                                                                                           26.1242196165329 31.0782662396643 22.8243477440400 22.3280126191681 43.3631035986013
                            37.7281540797592 24.0872182145107 38.2708765392885 37.6771723529620 26.1242196165329
                                                                                                                                                                                                                                                                                                                                                                                                                                                           23.3065605009871 22.4081778584554 44.9808092396814 39.1556508953255
                                                                                                                                                                                                                                                                                                                                                                                                                   0.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              41.9486260755836 37.9839915650424 15.9219850266585
                            14.6040438372539 26.2715370902888 30.5459618613093 14.5490447454510
                                                                                                                                                                                                                                                                                                                      31.0782662396643 23.3065605009871
                             56.3392870790292 25.3099446428611 40.9246789340187 56.2805781764249 22.8243477440400 22.4081778584554 41.9486260755836
                            43.0740876362725 \\ 21.6240433753013 \\ 7.50425937530314 \\ 43.0214450073284 \\ 22.3280126191681 \\ 44.9808092396814 \\ 37.9839915650424 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.9737207505211 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.973720750521 \\ 44.97372075052
                            1.77033377870531 \\ \phantom{0} 38.3945439775925 \\ \phantom{0} 36.2613595307519 \\ \phantom{0} 1.79049231777185 \\ \phantom{0} 43.631035986013 \\ \phantom{0} 39.1556508953255 \\ \phantom{0} 15.9219850266585 \\ \phantom{0} 57.4316496116755 \\ \phantom{0} 42.6705948109208 \\ \phantom{0} 10.661675 \\ \phantom{0} 10.661675
```

Now we can construct a new weight matrix  $\mathit{WM}$  in which the weight is doubled whenever the cities connected are in distinct countries.

```
> WM := Matrix(upperbound(DM), (i, j) ifelse(AfricanCities[i, 3] = AfricanCities[j, 3], DM[i, j], 2 \cdot DM[i, j]), datatype = double)
```

```
WM :=
                                                                                                                                      75.7496322348421
                                                                                                                                                                                                                               32.1700586745084 \quad 75.6251818647224 \quad 9.94565065667863 \quad 48.1744364290214 \quad 52.5430741805776 \quad 50.6198892857221 \quad 43.2480867506027 \quad 76.7890879551850 \quad 76.789087951850 \quad 76.7890851850 \quad 76.7890851850 \quad 76.7890851850 \quad 76.7890851850 \quad 76.
                                                                                                                                   32.1700586745084
                                                                                                                                                                                                                                                                                                                                                                                                                                    36.3822204721286 76.5417530785769 61.0919237226187 81.8493578680375 15.0085187506063 72.5227190615038
                                     72.9980993770630
                                                                                                                                                                                                                                                                                                                                      72.8871907089374
                                0.0622687580171630 75.6251818647224 72.8871907089374
                                                                                                                                                                                                                                                                                                                                                                                                                                       85.6952823243158 9.94565065667863 36.3822204721286 85.5708317433371
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    52.2484392330657 62.1565324793286 45.6486954880801 44.6560252383362 86.7262071972025
                                                                                                                                                                                                                                                                                                                                 75.3543447059240 52.2484392330657
                                   75 4563081595184 48 1744364290214 76 5417530785769
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  0
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               46.6131210019742 44.8163557169109 89.9616184793627 78.3113017906510
                                   29.2080876745078 52.5430741805776 61.0919237226187 29.0980894909020 62.1565324793286 46.6131210019742
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               0.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        83.8972521511673 75.9679831300847 31.8439700533169
                                   112.678574158058 50.6198892857221 81.8493578680375 112.561156352850 45.6486954880801 44.8163557169109 83.8972521511673
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   89.9474415011042 114.863299223351
                                   86.1481752725450 \\ \phantom{86.1481752725450} 43.2480867506027 \\ \phantom{86.1481752725450} 15.0085187506063 \\ \phantom{86.1481752725450} 86.0428900146567 \\ \phantom{86.1481752725450} 44.6560252383362 \\ \phantom{86.1481752725450} 89.9616184793627 \\ \phantom{86.1481752725450} 75.9679831300847 \\ \phantom{86.1481752725450} 89.961618479362 \\ \phantom{86.1481752725450} 75.9679831300847 \\ \phantom{86.1481752725450} 89.96161847 \\ \phantom{86.1481752725450} 75.9679831300847 \\ \phantom{86.148175272540} 75.9679811000000000000000000000
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      85.3411896218415
                                   1.77033377870531 \\ \phantom{0}76.7890879551850 \\ \phantom{0}72.5227190615038 \\ \phantom{0}1.79049231777185 \\ \phantom{0}86.7262071972025 \\ \phantom{0}78.3113017906510 \\ \phantom{0}31.8439700533169 \\ \phantom{0}31.8439700533169 \\ \phantom{0}31.863299223351 \\ \phantom{0}85.3411896218415 \\ \phantom{0}96.7262071972025 \\ \phantom{0}78.3113017906510 \\ \phantom{0}31.8439700533169 \\ \phantom{0}31.843970053169 \\ \phantom{0}31.843970053169 \\ \phantom{0}
```

We can now pass the weight matrix WM directly to TravelingSalesman to get a new tour.

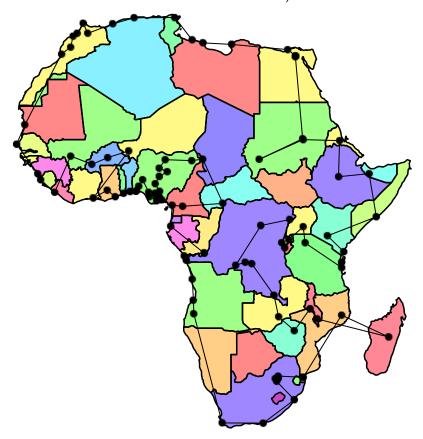
```
> W2, T2 := TravelingSalesman( G, WM, startvertex = "Cairo")
W2, T2 := 616.778513706285, ["Cairo", "Shubra el-Kheima", "Alexandria", "Benghazi", "Misratah",
    "Tripoli", "Tunis", "Algiers", "Oran", "Fez", "Tangier", "Rabat", "Casablanca", "Marrakesh",
    "Agadir", "Nouakchott", "Dakar", "Conakry", "Freetown", "Monrovia", "Bamako",
    "Bobo Dioulasso", "Ouagadougou", "Niamey", "Abidjan", "Kumasi", "Accra", "Lomé",
    "Abomey-Calavi", "Lagos", "Ikorodu", "Ibadan", "Ilorin", "Benin City", "Warri", "Port Harcourt",
    "Aba", "Uyo", "Umuahia", "Owerri", "Nnewi", "Onitsha", "Enugu", "Lokoja", "Abuja", "Jos",
    "Kaduna", "Kano", "Maiduguri", "N'Djamena", "Bangui", "Yaoundé", "Douala", "Libreville",
    "Kinshasa", "Brazzaville", "Pointe-Noire", "Cabinda", "Luanda", "Benguela", "Lubango",
    "Cape Town", "Gqeberha (Port Elizabeth)", "Durban (eThekwini)", "Vereeniging (Emfuleni)",
    "West Rand", "Johannesburg", "East Rand (Ekurhuleni)", "Pretoria (Tshwane)", "Matola",
    "Maputo", "Nampula", "Antananarivo", "Blantyre", "Lilongwe", "Harare", "Lusaka", "Lubumbashi",
    "Mbuji-Mayi", "Kananga", "Tshikapa", "Kisangani", "Bunia", "Bukavu", "Bujumbura", "Kigali",
    "Kampala", "Mwanza", "Dar es Salaam", "Zanzibar", "Mombasa", "Nairobi", "Mogadishu",
    "Hargeisa", "Addis Ababa", "Asmara", "Khartoum", "Nyala", "Omdurman", "Giza", "Cairo"]
```

> TG2 := Subgraph(G, Trail(T2))

 $TG2 := Graph \ 3$ : an undirected graph with 100 vertices and 100 edge(s)

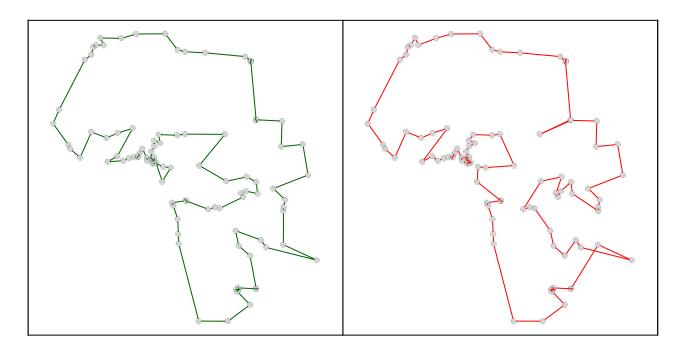
This new tour is similar to the previous one in many respects, but notably does not visit Sudan or the Democratic Republic of the Congo twice as the first one did, illustrating the effect of our increased edge weights.

> plots:-display(Import("example/AfricaMap.kml", base = datadir, title = none), DrawGraph(TG2, stylesheet = [vertexcolor = black]))



Finally we can draw the original tour (in green) and the new tour (in red) to see the effect of the altered weights.

>  $plots:-display(\langle DrawGraph(TG, stylesheet = [edgecolor = "darkgreen"])|DrawGraph(TG2, stylesheet = [edgecolor = red])\rangle)$ 



## Multigraph support

GraphTheory now supports multigraphs. That is, graphs in which there may be multiple edges between the same pair of vertices.

> 
$$MG := Graph \left( 4, \begin{bmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix}, multigraph \right)$$

MG := Graph 3: an undirected multigraph with 4 vertices and 5 edge(s)

The new command <u>IsMultigraph</u> tests whether a graph is a multigraph.

> IsMultigraph(MG)

true

Many other commands have been updated to support multigraphs.

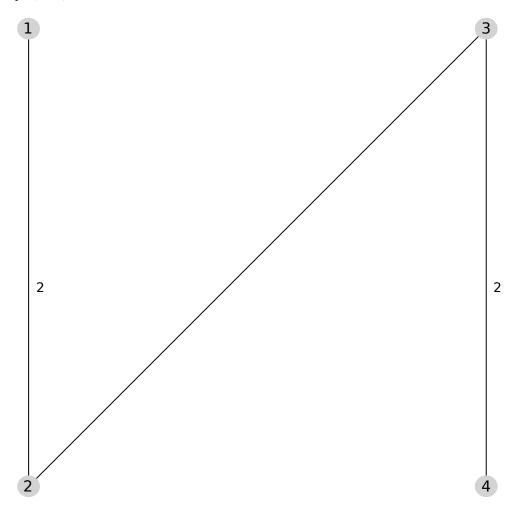
The <u>Edges</u> command for this graph returns a list, not a set, and repeats the edge between vertices 3 and 4 twice.

> Edges(MG)

$$[\{1,2\},\{1,2\},\{2,3\},\{3,4\},\{3,4\}]$$

Graph visualization commands such as DrawGraph will draw an integer weight on edges for which the edge multiplicity is greater than 1.

> DrawGraph(MG)



## **Graph products**

A graph product is a binary operation on graphs which takes two graphs  $G_1$  and  $G_2$  and produces a graph H with the following properties:

• The vertex set of H is the Cartesian product  $V(G_1) \times V(G_2)$  where  $V(G_1)$  and  $V(G_2)$  are the vertex sets of  $G_1$  and  $G_2$ , respectively.

$$V(G_l)$$

$$V(G_2)$$

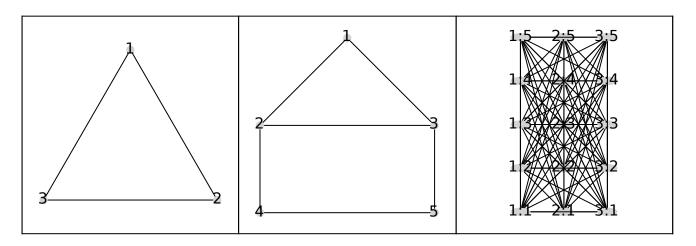
• Two vertices  $(u_1, v_1)$  and  $(u_2, v_2)$  of H are connected by an edge, iff some condition about  $u_1, v_1 \in G_1$  and  $u_2, v_2 \in G_2$  is fulfilled.

Many different graph products have been defined which differ in the condition imposed on the edges. To the existing <u>CartesianProduct</u> and <u>TensorProduct</u> commands, the following new graph products have been added:

Name	Edge Condition		
ConormalProduct	$u_1$ and $v_1$ share an edge in $G_1$ or $u_2$ and $v_2$ share an edge in $G_2$		
LexicographicProduc	$u_1$ and $v_1$ share an edge in $G_1$ , or $u_1 = v_1$ in $G_1$ and $u_2$ and $v_2$		
<u>t</u>	share an edge in ${\it G}_{2}$		
	$u_{_1}$ and $v_{_1}$ share an edge in $G_{_1}$ and $u_{_2}$ and $v_{_2}$ share an edge in		
ModularProduct	$G_{2}$ ,		
	or $u_{_1}$ and $v_{_1}$ do not share an edge in $G_{_1}$ and $u_{_2}$ and $v_{_2}$ do not		
	share an edge in ${\cal G}_{\!_2}$		
	$u_1 = v_1$ in $G_1$ and $u_2$ and $v_2$ share an edge in $G_2$ ,		
<u>StrongProduct</u>	or $u_1$ and $v_1$ share an edge in $G_1$ and $u_2 = v_2$ in $G_2$		
	or $u_1$ and $v_1$ share an edge in $G_1$ and $u_2$ and $v_2$ share an edge in		
	$G_{2}$		

- $\succ C := CycleGraph(3)$ 
  - $C := Graph \ 4$ : an undirected graph with 3 vertices and 3 edge(s)
- $\rightarrow$  H := SpecialGraphs:-HouseGraph()
  - H := Graph 5: an undirected graph with 5 vertices and 6 edge(s)
- $\rightarrow LP := LexicographicProduct(C, H)$ 
  - $LP := Graph \ 6$ : an undirected graph with 15 vertices and 93 edge(s)

## $\rightarrow$ plots:-display(DrawGraph~( $\langle C|H|LP\rangle$ ))



## **Additions to SpecialGraphs**

Maple 2023 provides support for 6 additional Special Graphs, bringing the total to 119.

> with(SpecialGraphs):

Bishops's Graph	<b>Bouquet Graph</b>	Dipole Graph
> BG46 :=     BishopsGraph(4, 5)  BG46 :=     Graph 7: an undirected     graph with 20 vertices and     40 edge(s)  > DrawGraph(BG46,     stylesheet     = [vertexpadding     = 12])  1:5    2:5    3:5    4:5  1:4    2x4    3x4    4:4  1:3    2x3    3x3    4:3  1:2    2x2    3x2    4:2  1:1    2:1    3:1    4:1	> BG := BouquetGraph(5) BG := Graph 8: an undirected multigraph with 1 vertex, no edges, and 5 self-loops > DrawGraph(BG, stylesheet = [vertexpadding = 12])  5 1	> DG := DipoleGraph(2) DG := Graph 9: an undirected multigraph with 2 vertices and 2 edge(s) > DrawGraph(DG, stylesheet = [vertexpadding = 12])  1 2

<b>Hamming Graph</b>	House Graph	Windmill Graph
> HG33 :=     HammingGraph(3,     3)  HG33 :=     Graph 10: an undirected     graph with 27 vertices and     81 edge(s)  > DrawGraph(HG33, style     = spring, stylesheet     = [vertexpadding     = 12])	> HG := HouseGraph()  HG :=  Graph 11: an undirected graph with 5 vertices and 6 edge(s)  > DrawGraph(HG, stylesheet = [vertexpadding = 10])	> WG := WindmillGraph(3,7) WG := Graph 12: an undirected graph with 15 vertices and 21 edge(s) > DrawGraph(WG, stylesheet = [vertexpadding = 10])  6-7 4 5 8 11 10 13 12