

Mathematica: funzioni per grafi e reti

Sintesi delle funzioni principali per la costruzione,
l'elaborazione e la visualizzazione di grafi con il
software Mathematica ver. 11

Corso di Ottimizzazione Combinatoria (IN440)
a.a. 2017-2018

Grafo, vertici, spigoli

Creazione e modifica di un grafo attraverso l'aggiunta e la rimozione di vertici e spigoli

```
G = Graph[{1<->2, 2<->3, 3<->4, 2<->4}]  
  
G = VertexAdd[G, 5]  
G = VertexAdd[G, {6, 7}]  
G = VertexDelete[G, 6]  
  
G= EdgeAdd[G, {7<->8, 2<->8}]  
G = EdgeDelete[G, 2<->3]  
  
VertexList[G]  
EdgeList[G]  
  
AdjacencyList[G, 1]  
Export["grafo.dat", AdjacencyMatrix[G]]  
  
G = AdjacencyGraph[{{0,1,0}, {0,0,1}, {1,0,0}}]  
AdjacencyMatrix[G] // MatrixForm
```

Visualizzazione di grafi

Visualizzazione di grafi, esportazione e importazione di grafi su file (in formato grafico non importabile o in formato “dati”, modificabile e importabile)

```
GraphPlot[G, VertexLabeling->True]
```

```
Export["grafo.gif", G]
```

```
Export["grafo.graphml", G]
```

```
H = Import["grafo.graphml"]
```

```
Export["grafo.gv", G]
```

```
H = Import["grafo.gv"]
```

Grafi notevoli

Creazione di grafi a partire da un grafo “notevole” definito sulla base di un parametro:

```
G = CompleteGraph[5]
```

```
G = CompleteGraph[{4,3}]
```

```
G = RandomGraph[{10,7}]
```

```
G = CycleGraph[17]
```

```
G = StarGraph[9]
```

```
G = WheelGraph[9]
```

```
PlanarGraphQ[G]
```

Proprietà di un grafo

Calcolo di alcune proprietà di un grafo (le istruzioni che terminano con la lettera "Q" sono delle *"query"*, delle domande che ammettono come risposta "vero" o "falso")

```
GraphDiameter[G]  
GraphCenter[G]  
GraphRadius[G]  
ConnectedGraphQ[G]  
EulerianGraphQ[G]  
HamiltonianGraphQ[G]  
FindHamiltonianCycle[G]  
HighlightGraph[G, FindHamiltonianCycle[G], GraphHighlightStyle->"Thick"]  
  
FindClique[G, Infinity]  
HighlightGraph[G, FindClique[G,Infinity],GraphHighlightStyle->"Thick"]
```

Spanning Tree di un Grafo

Calcolo di un albero ricoprente di un grafo:

```
FindSpanningTree[G]  
  
HighlightGraph[G, FindSpanningTree[G], GraphHighlightStyle -> "Thick"]  
  
GraphPlot[ HighlightGraph[G, FindSpanningTree[G], GraphHighlightStyle ->  
"Thick"], VertexLabeling -> True]  
  
Export["spanningTree.gif", HighlightGraph[G, FindSpanningTree[G],  
GraphHighlightStyle->"Thick"]]
```