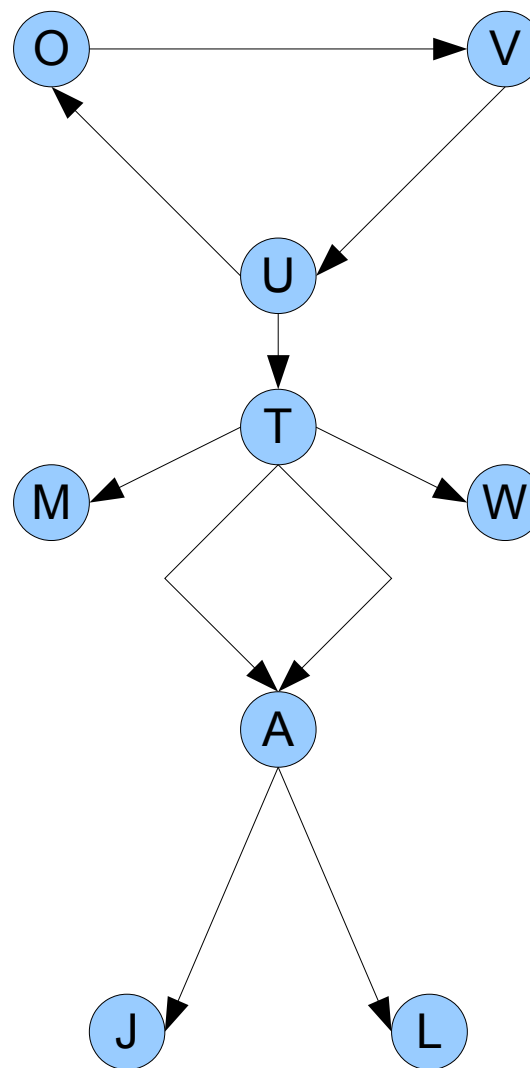


# Programmeerimise süvendatud algkursus ITI0140

2014

# Graaf

- Graaf on konstruktsioon, millega saab kirjeldada objektidevahelisi seoseid.
- Graaf koosneb tippudest ja tippude vahelistest servadest.

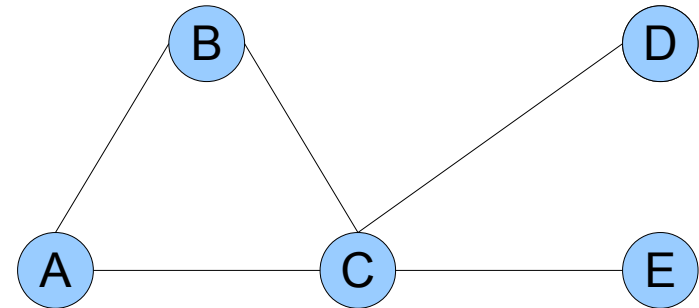


# Graafide näiteid

- Sugupuu
  - kes on kelle järglane
- Linnadevaheline kauguste kaart
  - kust ja kuhu saab
- Ühistranspordivõrk
  - kust, kuhu, millega ja mis hinnaga saab
- Arvutivõrgu topoloogia
  - mis seadmed on omavahel seotud
- Tutvuste võrgustik
  - kes, keda ja kelle kaudu tunneb
  - [http://en.wikipedia.org/wiki/Six\\_Degrees\\_of\\_Kevin\\_Bacon](http://en.wikipedia.org/wiki/Six_Degrees_of_Kevin_Bacon)
- Olekuautomaat
  - visualiseeritud regulaaravaldis

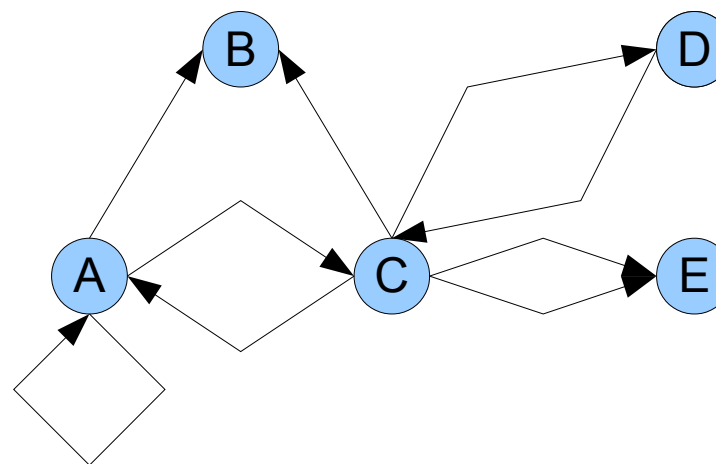
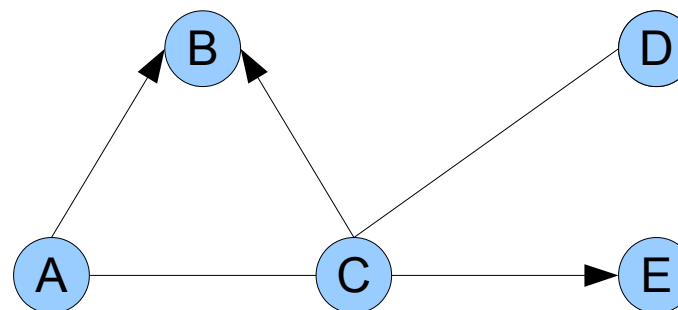
# Lihtgraaf

- Graaf koosneb tippudest
  - A, B, C, D, E
- Tippude vahel on servad
  - AB, AC, BC, CD, CE
- Servaga seotud tipud on naabertipud



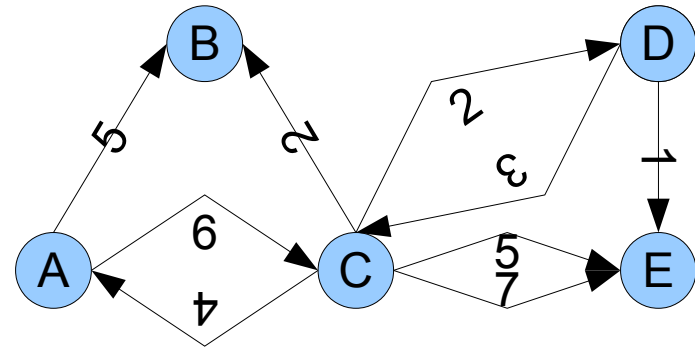
# Suunatud graaf ja multigraaf

- Suunatud servi nimetatakse kaarteks või noolteks
- Suunatud servadega graafi nimetatakse suunatud e. orienteeritud graafiks
- Suunamata servad on mõlemasuunalised
  - AB, AC, CA, CB, CD, CE, DC
- Kui tippude vahel on mitu serva, nimetatakse graafi multigraafiks
  - AA, AB, AC, CA, CB, CD, CE, CE, DC



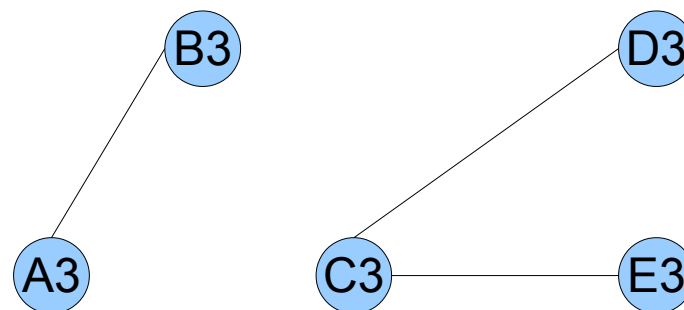
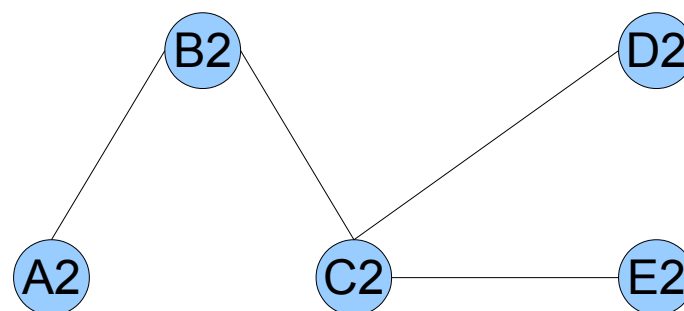
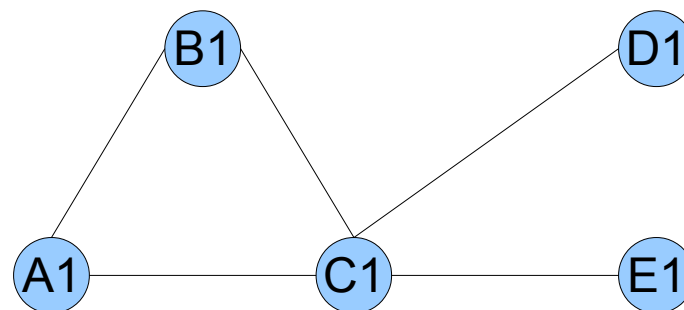
# Kaalutud graaf

- Graafi, mille seoste on omistatud väärtused, nimetatakse kaalutud graafiks
- Servade e. naabertippude jada kahe tipu vahel nimetatakse teeks
  - $A \rightarrow E$ : AC, CD, DE
- Lühim tee, kahe tipu vahel on kaugus
  - $A \rightarrow E$ : AC, CE
- Kaugus võib arvestada ka kaalusid
  - AC, CE =  $6 + 5 = 11$
  - AC, CD, DE =  $6 + 2 + 1 = 9$



# Seotud ja mitteseotud graaf

- Seotud graafis leidub iga kahe tipu vahel tee
- Seotud graafi eriliik on täielikult hargnev graaf e. puu
- Graafi mittesidusaid osasid nimetatakse komponentideks



# Näiteid eriomadustega graafidest

- Täisgraaf (või täielik graaf)
  - kõik tipud on omavahel naabertipud (iga kahe tipu vahel on serv)
- Tühigraaf
  - servade ja kaarteta graaf
- Euleri graaf
  - leidub tee, mis läbib kõiki servi üks kord ja moodustab tsükli
- Hamiltoni graaf
  - leidub tee, mis läbib kõiki tippe üks kord ja moodustab tsükli
- Peterseni graaf (10 tippu, 15 serva, igal tipul on kolm serva)



# NetworkX

- Pythoni moodul, mis võimaldab graafe koostada, genereerida, joonistada ja rakendada mitmesuguseid graafialgoritme
  - <http://networkx.github.io>
- Dokumentatsioon:
  - <http://networkx.github.io/documentation.html>
- Näited
  - <http://networkx.github.io/documentation/latest/examples>
- Algoritmid
  - <http://networkx.github.io/documentation/latest/reference/algorithms.html>
  - [http://networkx.github.io/documentation/latest/reference/algorithms.shortest\\_paths.html](http://networkx.github.io/documentation/latest/reference/algorithms.shortest_paths.html)
- Joonistamine
  - <http://networkx.github.io/documentation/latest/examples/drawing/index.html>

# NetworkX mõned kasulikud näited

- Four Grids:  
[http://networkx.github.io/documentation/latest/examples/drawing/four\\_grids.html](http://networkx.github.io/documentation/latest/examples/drawing/four_grids.html)
- House With Colors:  
[http://networkx.github.io/documentation/latest/examples/drawing/house\\_with\\_colors.html](http://networkx.github.io/documentation/latest/examples/drawing/house_with_colors.html)
- Edge Colormap:  
[http://networkx.github.io/documentation/latest/examples/drawing/edge\\_colormap.html](http://networkx.github.io/documentation/latest/examples/drawing/edge_colormap.html)
- Node Colormap:  
[http://networkx.github.io/documentation/latest/examples/drawing/node\\_colormap.html](http://networkx.github.io/documentation/latest/examples/drawing/node_colormap.html)
- Weighted Graph:  
[http://networkx.github.io/documentation/latest/examples/drawing/weighted\\_graph.html](http://networkx.github.io/documentation/latest/examples/drawing/weighted_graph.html)
- Random Geometric Graph:  
[http://networkx.github.io/documentation/latest/examples/drawing/random\\_geometric\\_graph.html](http://networkx.github.io/documentation/latest/examples/drawing/random_geometric_graph.html)

# Ülesanne

- Kasutades NetworkX moodulit, genereerida mõnekümne tipuga mittetäielik graaf, leida mingi kahe tipu vahel lühim tee (kui mitu, siis võtta suvaline) ja visualiseerida graaf ning eristatavalt vastav lühim tee. Genereerimise algväärtustamiseks kasutada oma matriklit, ka alg- ja lõpptipu valiku jaoks kasutada mingil moel oma matriklit.
- Vihjeks
  - Juhusliku graafi genereerimine:  
[http://networkx.github.io/documentation/latest/reference/generated/networkx.generators.random\\_graphs.fast\\_gnp\\_random\\_graph.html#networkx.generators.random\\_graphs.fast\\_gnp\\_random\\_graph](http://networkx.github.io/documentation/latest/reference/generated/networkx.generators.random_graphs.fast_gnp_random_graph.html#networkx.generators.random_graphs.fast_gnp_random_graph)
  - Lühima tee leidmine:  
[http://networkx.github.io/documentation/latest/reference/generated/networkx.algorithms.shortest\\_paths.generic.shortest\\_path.html#networkx.algorithms.shortest\\_paths.generic.shortest\\_path](http://networkx.github.io/documentation/latest/reference/generated/networkx.algorithms.shortest_paths.generic.shortest_path.html#networkx.algorithms.shortest_paths.generic.shortest_path)
  - Graafi visualiseerimine:  
[http://networkx.github.io/documentation/latest/reference/drawing.html#module-networkx.drawing.nx\\_pylab](http://networkx.github.io/documentation/latest/reference/drawing.html#module-networkx.drawing.nx_pylab)