Lecturer: Andrew Parkes
http://www.cs.nott.ac.uk/~ajp/

G52ADS 2014-15
Minimum Spanning Trees

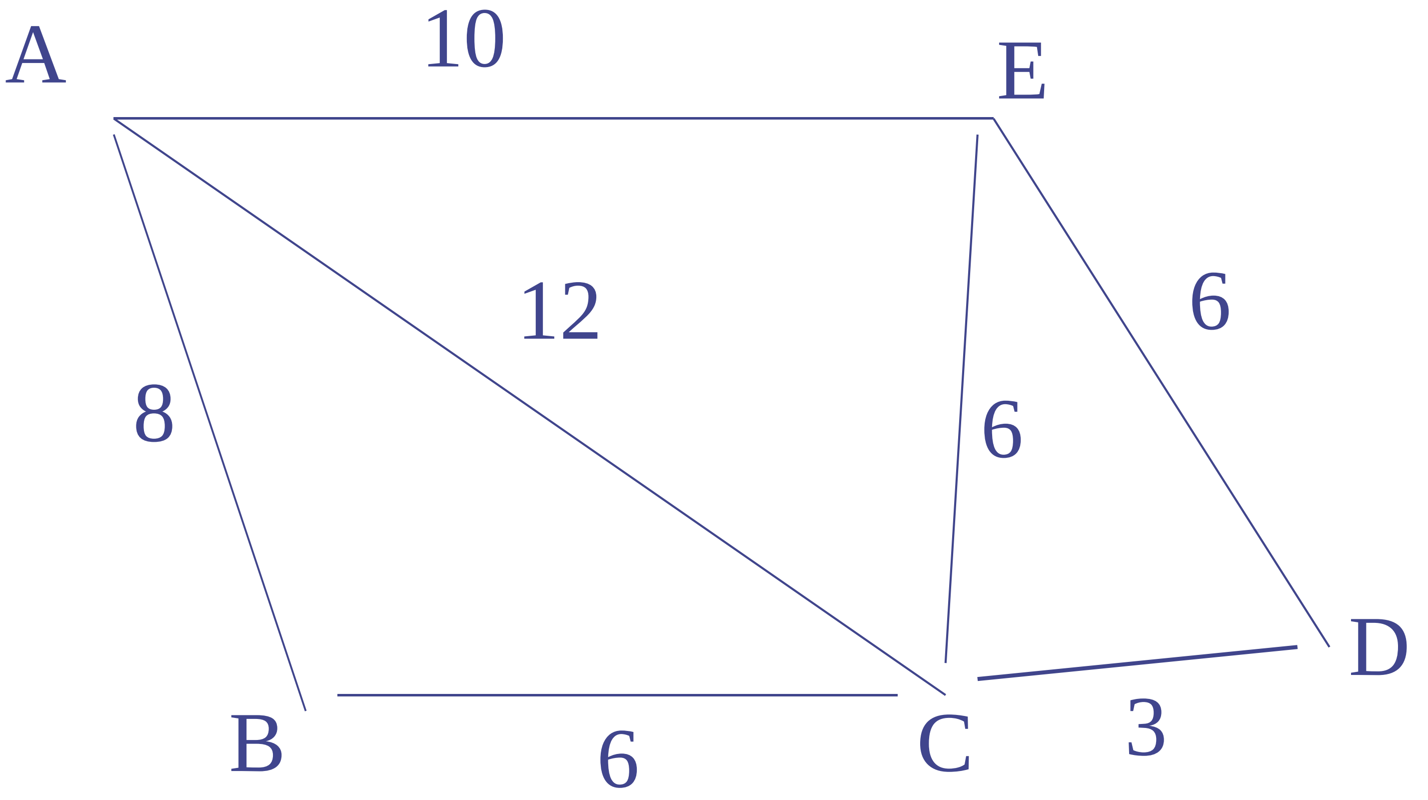
Spanning Tree

- Input: connected, undirected graph
- Output: a tree which connects all vertices in the graph using only the edges present in the graph

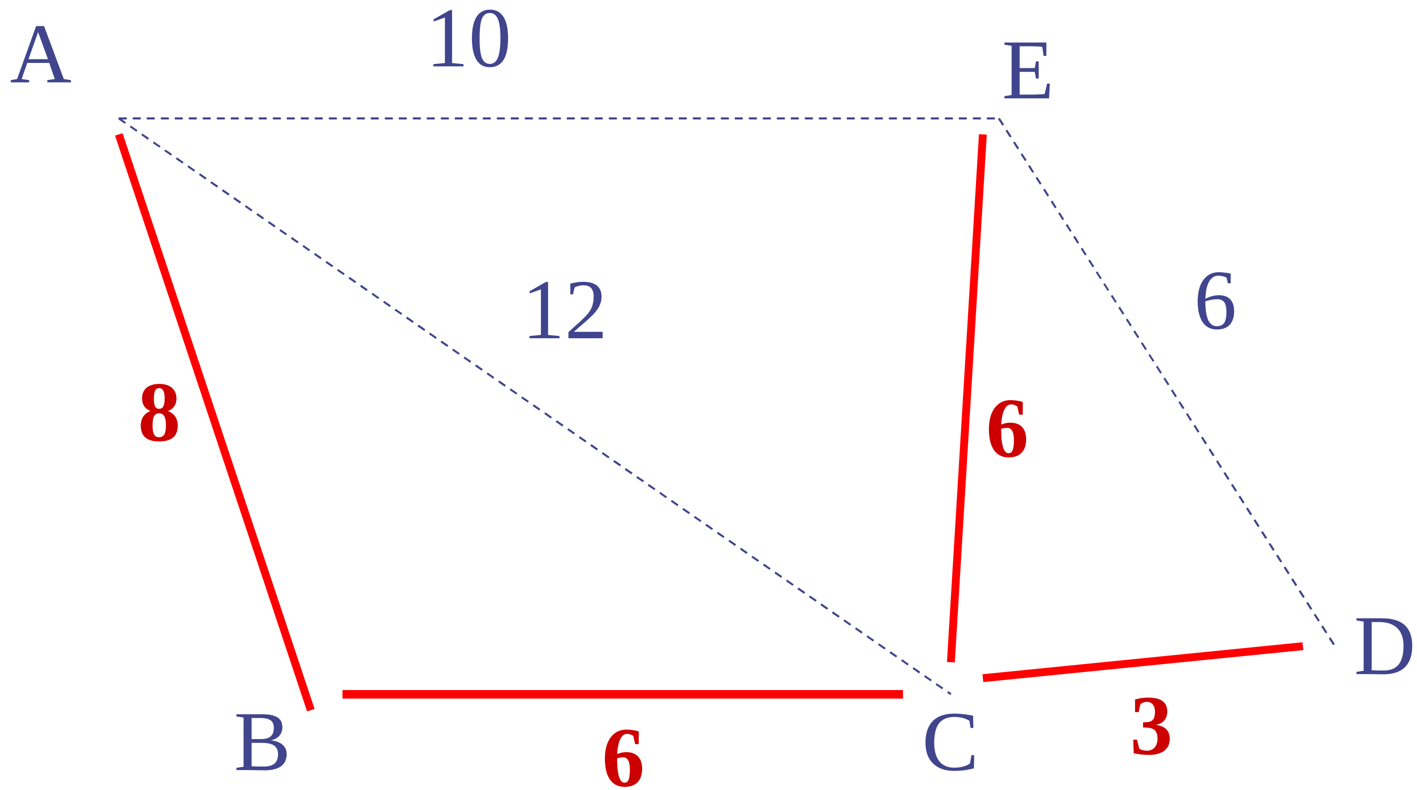
Minimum Spanning Tree

- Input: connected, undirected, weighted graph
- Output: a spanning tree
  - (connects all vertices in the graph using only the edges present in the graph)
  - and is minimum in the sense that the sum of weights of the edges is the smallest possible for any spanning tree

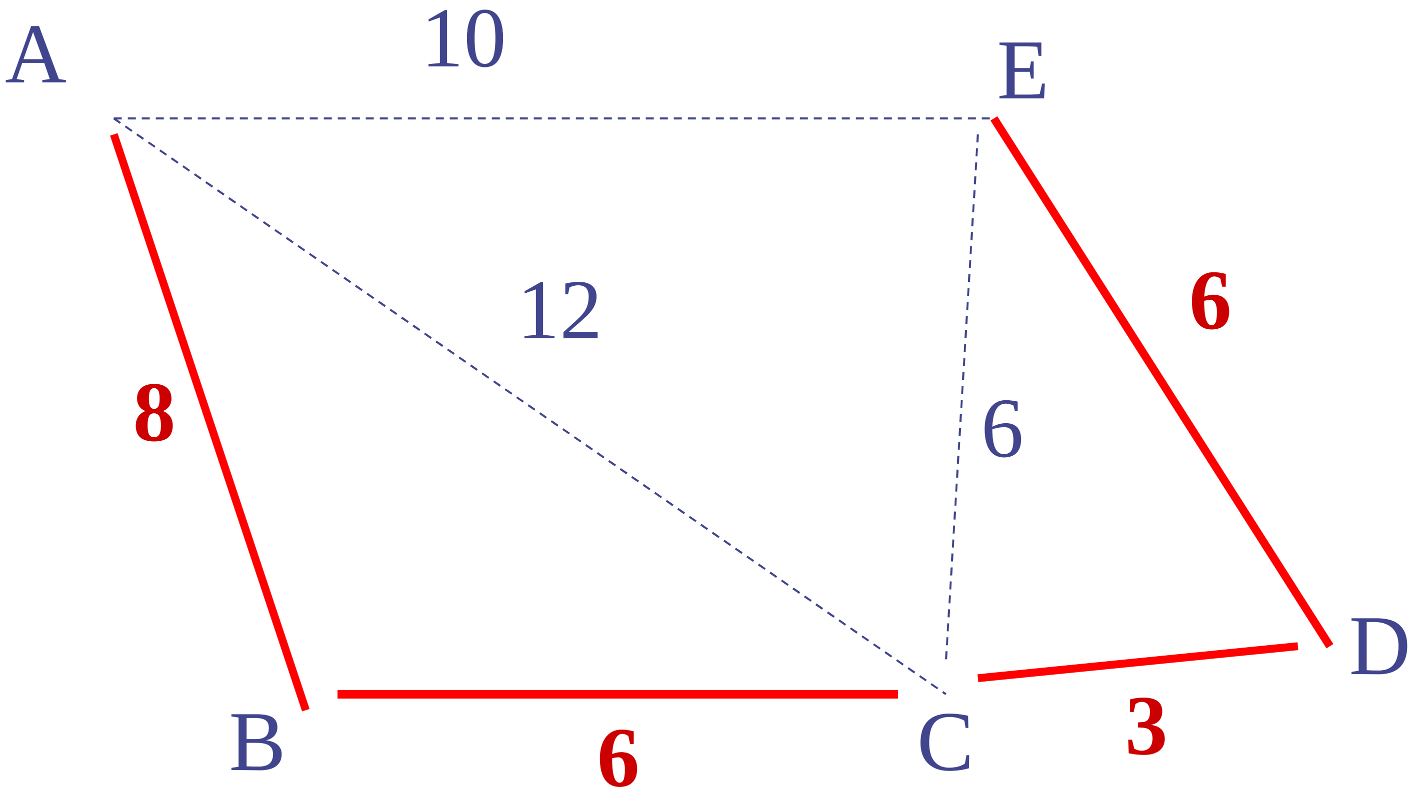
Example: graph



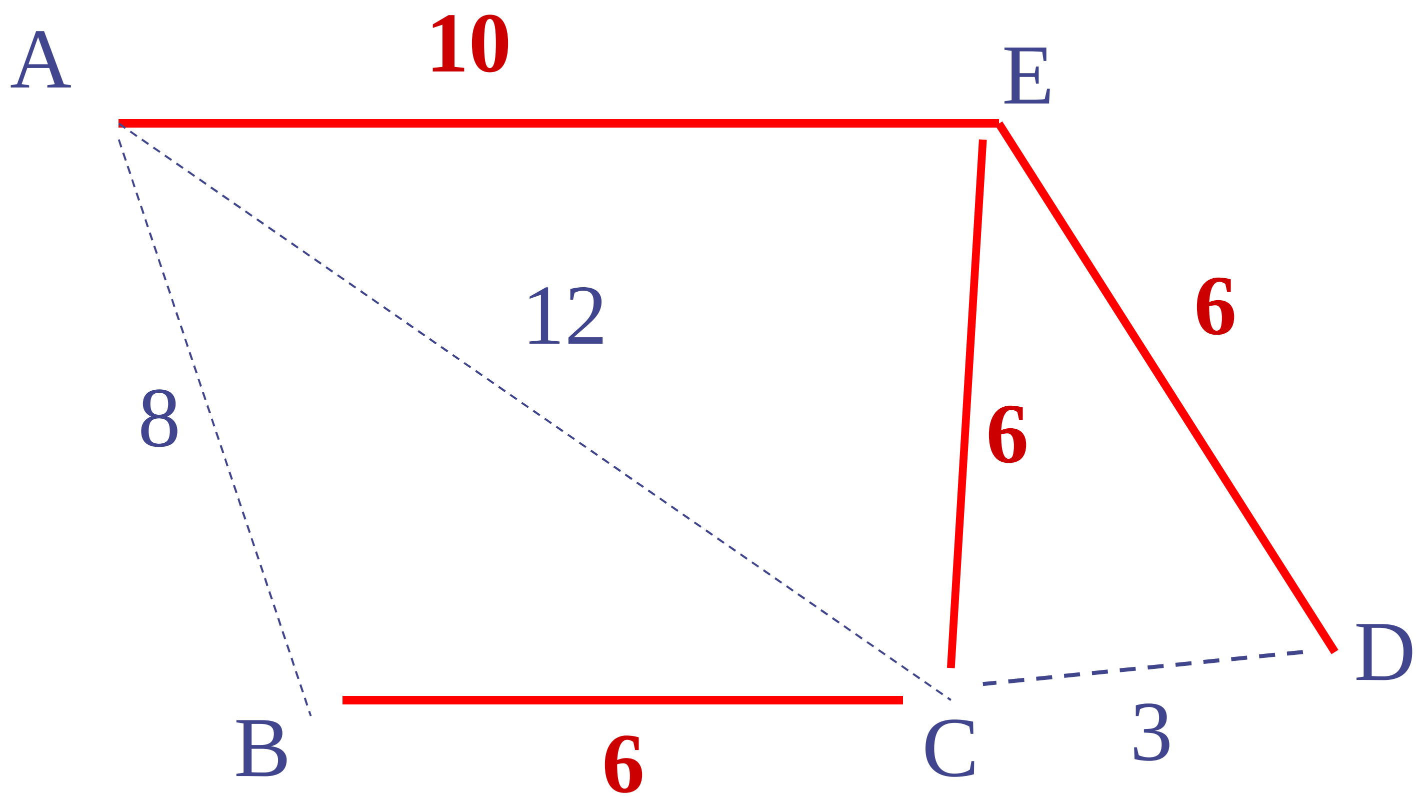
Example: MST (cost 23)



Example: another MST (cost 23)



Example: not MST (cost 28)



Why MST is a tree

- We really want a minimum spanning sub-graph
  - that is, a subset of the edges that is connected and that contains every node
- (Assuming all weights are non-negative)
  If the graph has a cycle then we can remove an edge of the cycle, and the graph will still be connected, and will have a smaller weight
- If an graph is connected and acyclic then it is a tree

An MST is a TREE

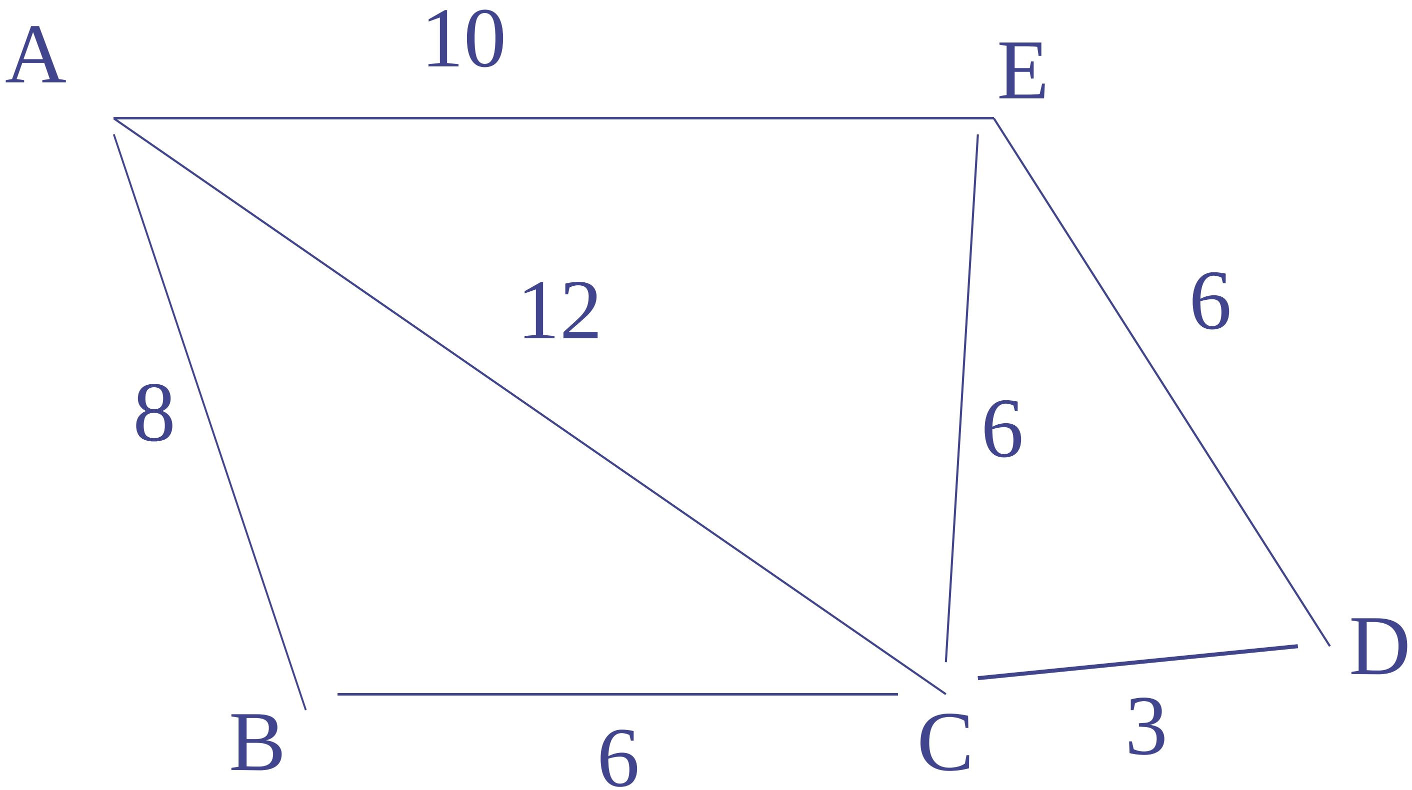
- Do not confuse a minimum **TREE** with a “minimum” (shortest) **PATH**
- Finding the shortest path that goes through all the nodes is a different problem from the MST (and much harder)
- (Many people confused these on last years exam).

Prim’s algorithm

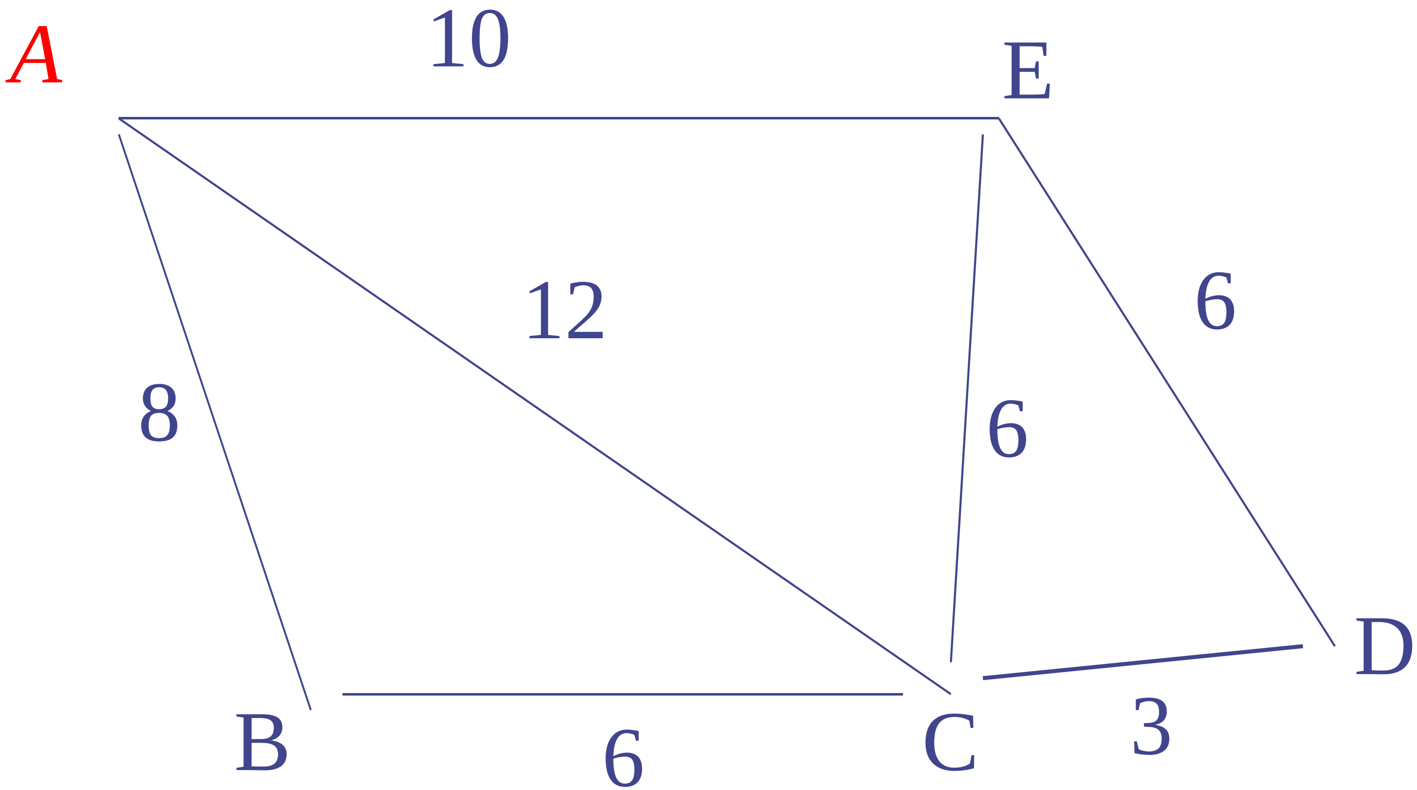
To construct an MST:

- Start by picking any vertex M
- Choose the shortest edge from M to any other vertex N
- Add edge (M,N) to the MST
- Loop:
  - Continue to add at every step the shortest edge from a vertex in MST to a vertex outside, until all vertices are in MST

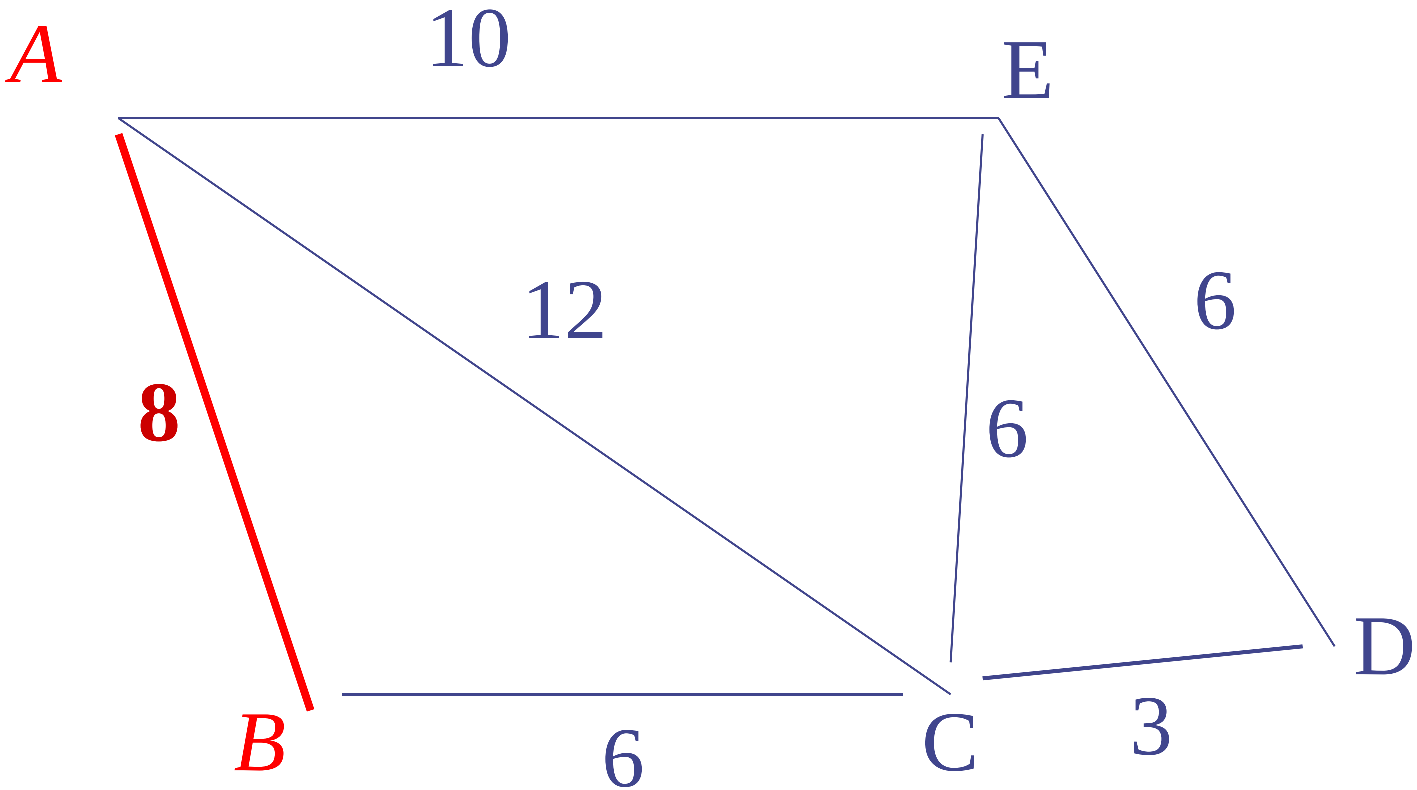
Example



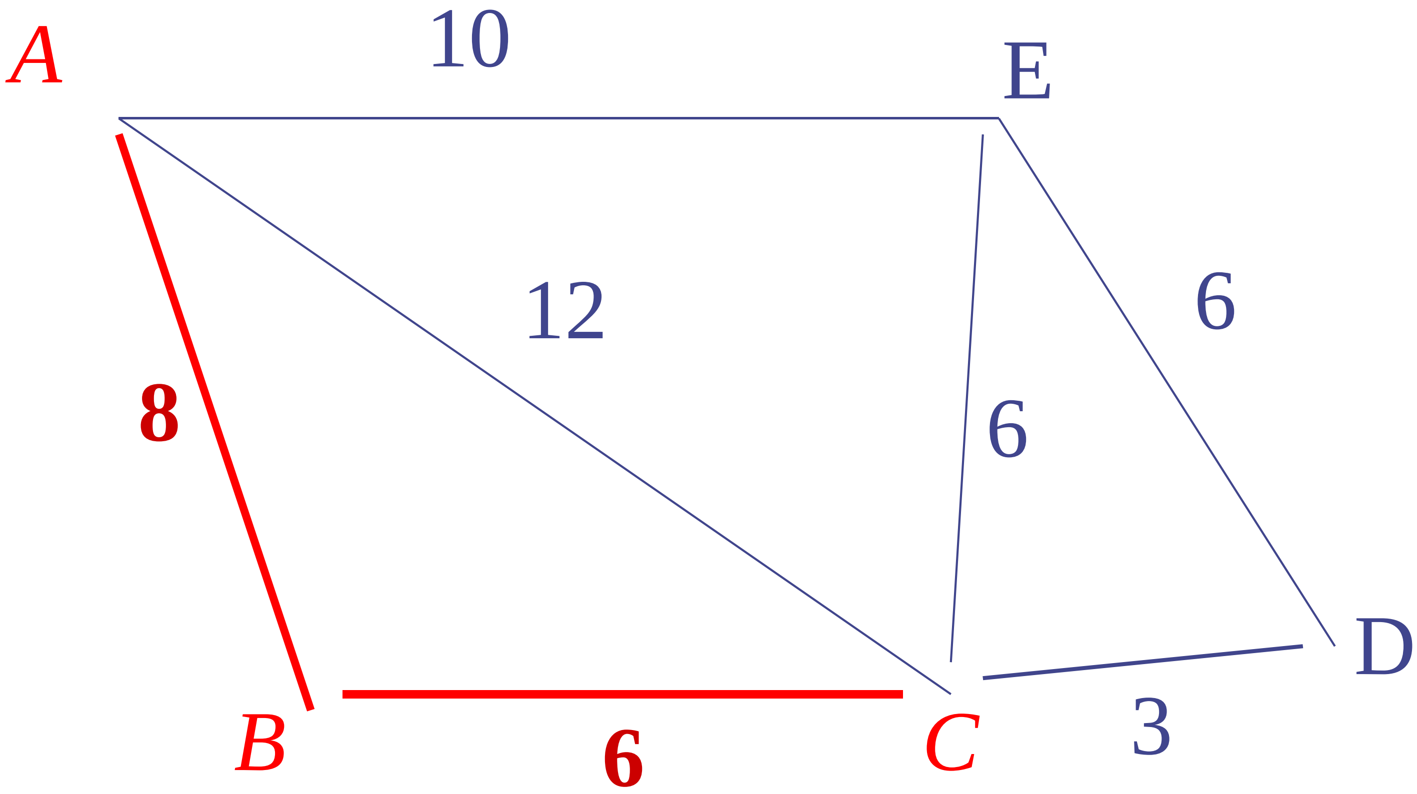
Example



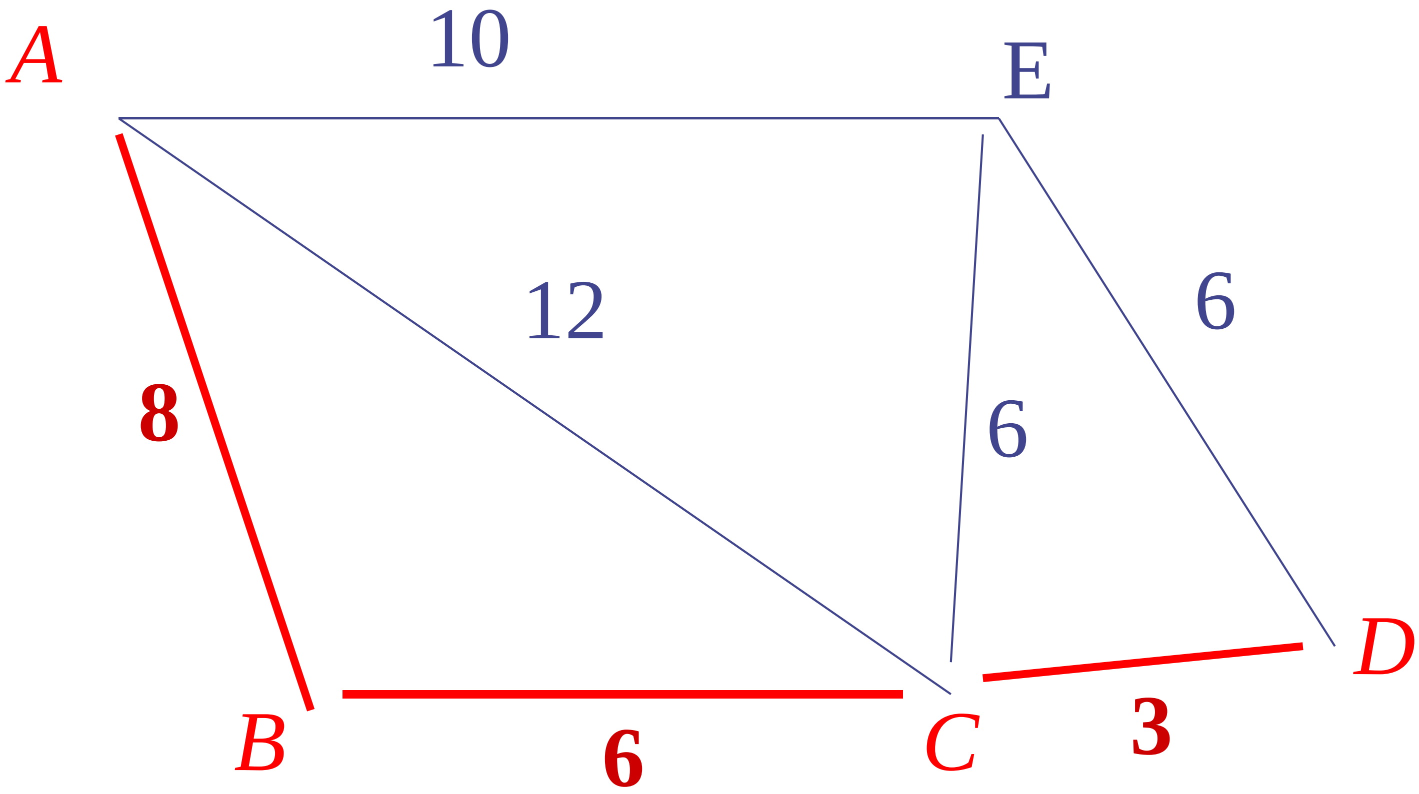
Example



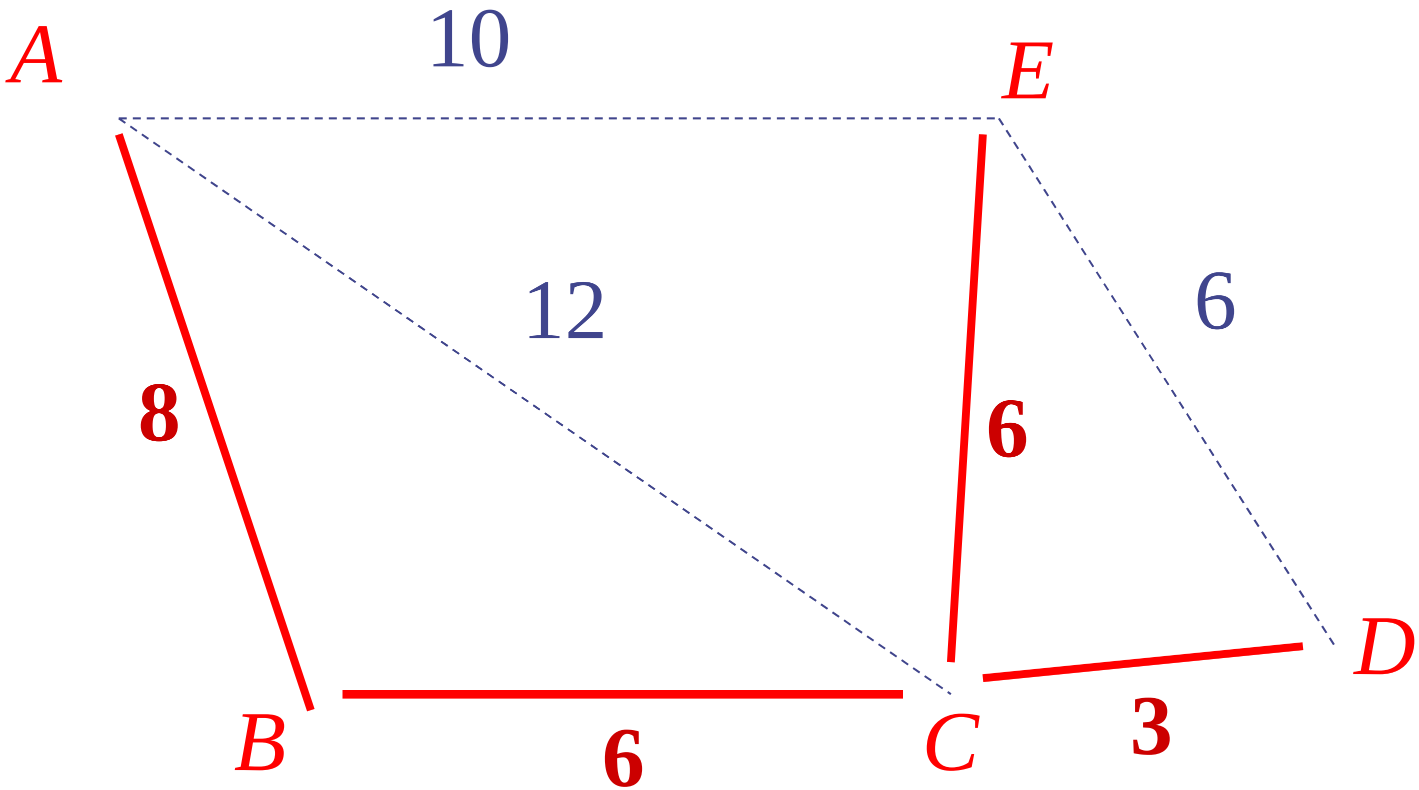
Example



Example



Example



Why is this optimal!?

- GoTa. Proposition 13.25 (Section 13.7)
- "Let G be a weighted connected graph, and
  - let V1 and V2 be a partition of the vertices of G into two disjoint non-empty sets.
  - Furthermore, let e be an edge with minimum weight from among those with one endpoint in V1 and the other in V2.
  - There is a MST that has e as one of its edges."

Justification of Prop. 13.25

- Argument by contradiction.
- Suppose that some minimum spanning tree T that is better than all trees containing e.
- Then can add edge e to T and remove some other edge between V1 and V2 and obtain a better MST

Prop 13.25 and Prims

- At each stage:
  - V1 = vertices within the current MST
  - V2 = “the rest” (vertices not in the MST)
  - The algorithm adds a minimum weight edge between V1 and V2, and so this edge must be part of some MST
  - Hence, the construction cannot make a “fatal mistake” – at no point can it add an edge not part of an MST

Greedy algorithm

- Prim's algorithm for constructing a Minimal Spanning Tree is a ***greedy algorithm***: it just adds the shortest edge without worrying about the overall structure, without looking ahead. It makes a locally optimal choice at each step.