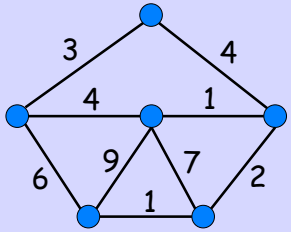
Mathematics for Computer Science
MIT 6.042J/18.062J

Minimum Weight Spanning Trees

Minimum Spanning Trees

Suppose edges have weights:



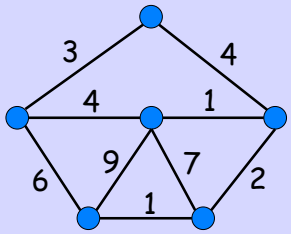
Find min weight spanning tree?

Build MST using gray edges

- Start with vertices, no edges
- Color components black & white
- Graph is connected, so have:
- gray edge ::= ○——●
- add min weight gray edge

Minimum Spanning Trees

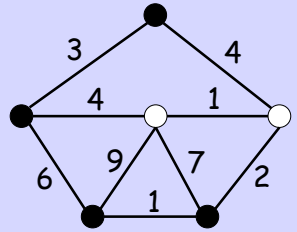
color components



initial components:
isolated vertices

Minimum Spanning Trees

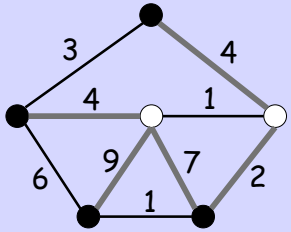
color components



initial components:
isolated vertices

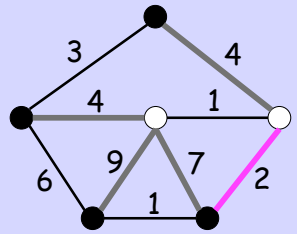
Minimum Spanning Trees

gray edges



Minimum Spanning Trees

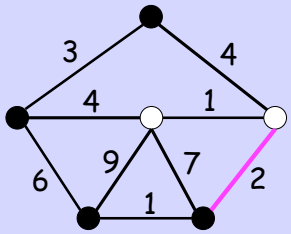
gray edges: min weight



repeat:

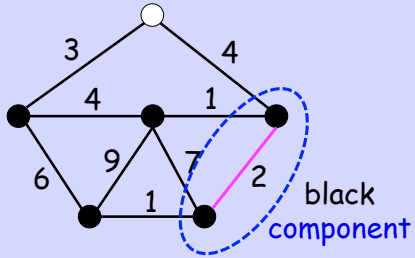
Minimum Spanning Trees

re-color components



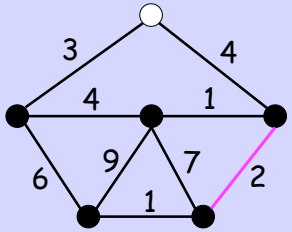
Minimum Spanning Trees

re-color components



Minimum Spanning Trees

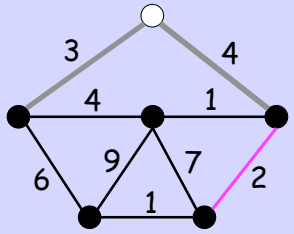
re-color components



so gray edges must differ from previously selected

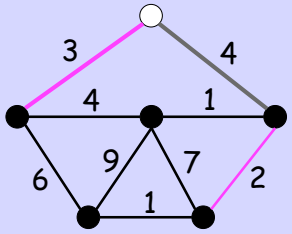
Minimum Spanning Trees

gray edges



Minimum Spanning Trees

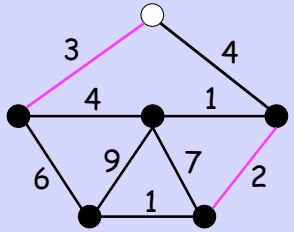
gray edges: min weight



repeat:

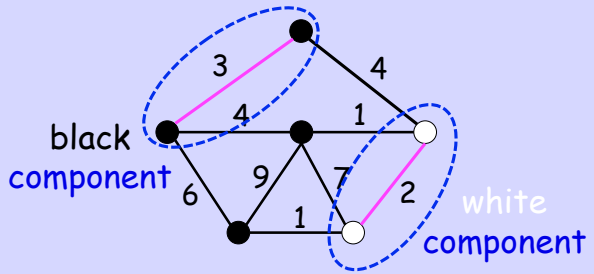
Minimum Spanning Trees

re-color components



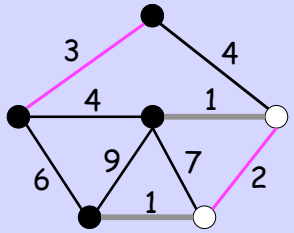
Minimum Spanning Trees

re-color components



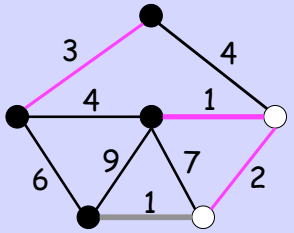
Minimum Spanning Trees

gray edges

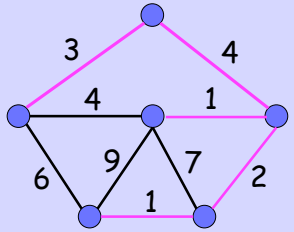


Minimum Spanning Trees

gray edges: min weight



Minimum Spanning Trees



etc.

Ways to grow an MST

- start at any vertex, keep building one tree. (Prim)
- keep choosing min weight edge between diff components (Kruskal)
- grow trees in parallel (Boruvka)
- All special cases of gray edges

Enough Gray Edges

We have shown

Theorem: Any connected graph has a spanning tree whose

edges are all min weight gray edges.

Still to prove:

any such tree is min-weight