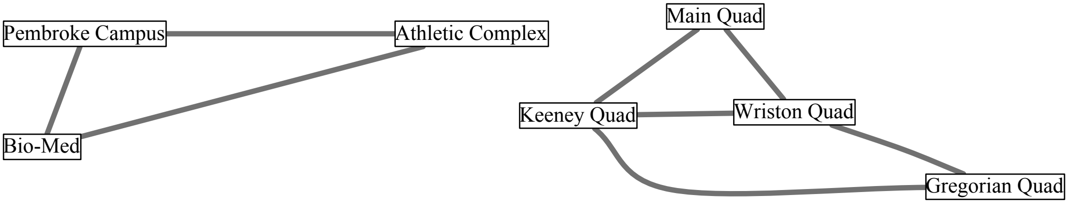
Minimum spanning forest



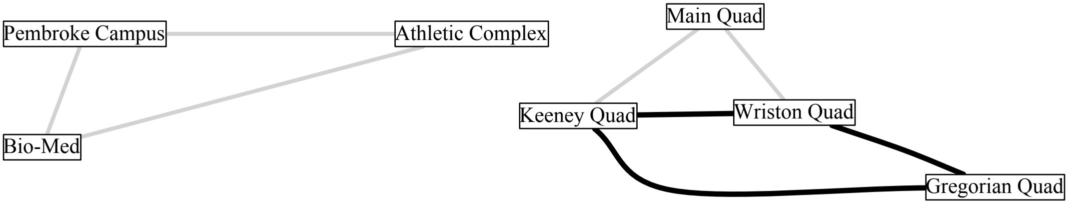
Definition: A sequence of edges $[\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \ldots, \{x_{k-1}, x_k\}]$ is called an x_1-to-x_k *path*.

Example "Main Quad"-to-"Gregorian Quad" paths in above graph:

- one goes through "Wriston Quad" ,
- one goes through "Keeney Quad"

Definition: A x-to-x path is called a *cycle*.

Minimum spanning forest



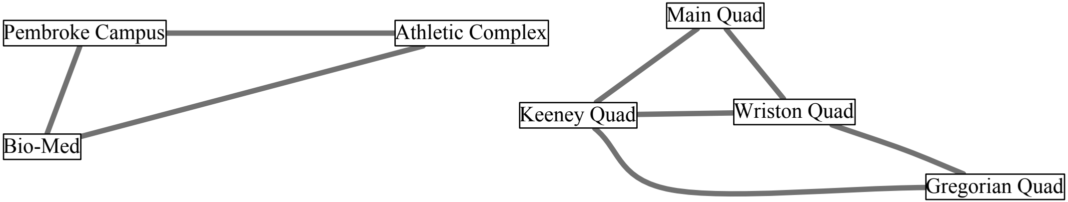
Definition: A sequence of edges $[\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \ldots, \{x_{k-1}, x_k\}]$ is called an x_1-to-x_k *path*.

Example "Main Quad"-to-"Gregorian Quad" paths in above graph:

- one goes through "Wriston Quad" ,
- one goes through "Keeney Quad"

Definition: A x-to-x path is called a *cycle*.

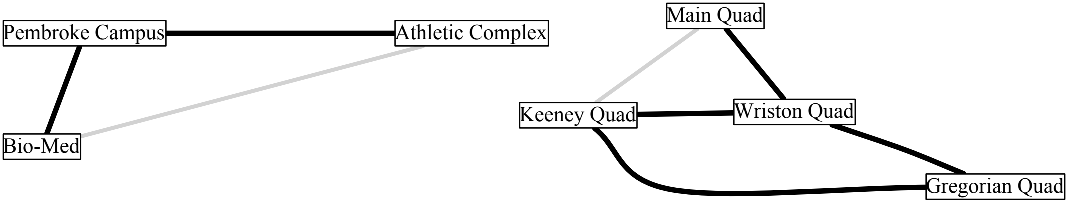
Minimum spanning forest: spanning



Definition: A set S of edges is *spanning* for a graph G if, for every edge $\{x, y\}$ of G, there is an x-to-y path consisting of edges of S.

Soon we see connection between this use of "spanning" and its use with vectors.

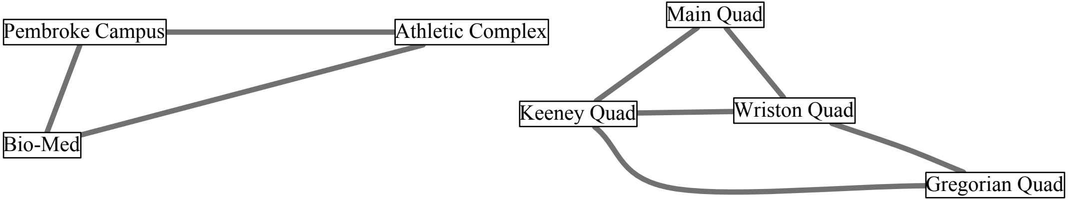
Minimum spanning forest: spanning



Definition: A set S of edges is *spanning* for a graph G if, for every edge $\{x, y\}$ of G, there is an x-to-y path consisting of edges of S.

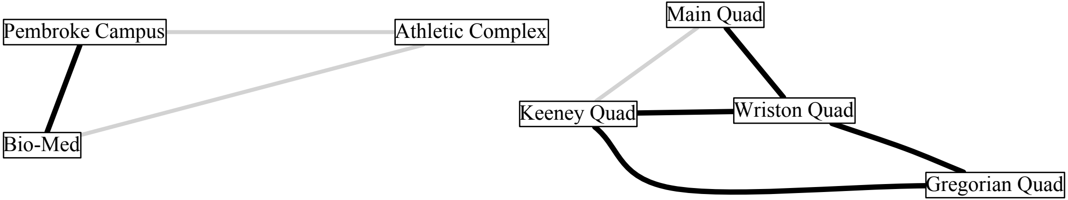
Soon we see connection between this use of "spanning" and its use with vectors.

Minimum spanning forest: forest



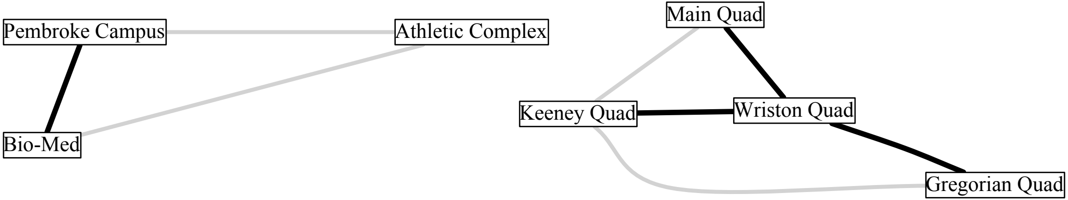
Definition: A set of edges of G is a *forest* if the set includes no cycles.

Minimum spanning forest: forest



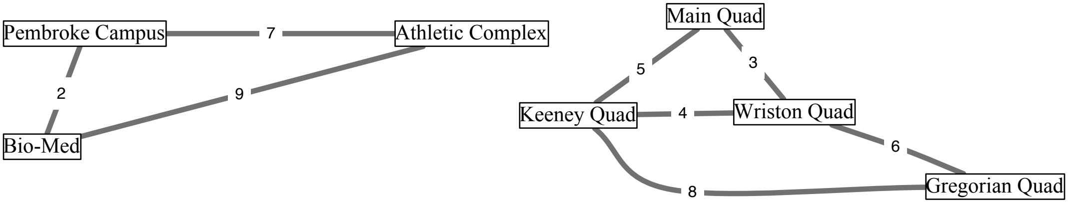
Definition: A set of edges of G is a *forest* if the set includes no cycles.

Minimum spanning forest: forest



Definition: A set of edges of G is a *forest* if the set includes no cycles.

Minimum spanning forest



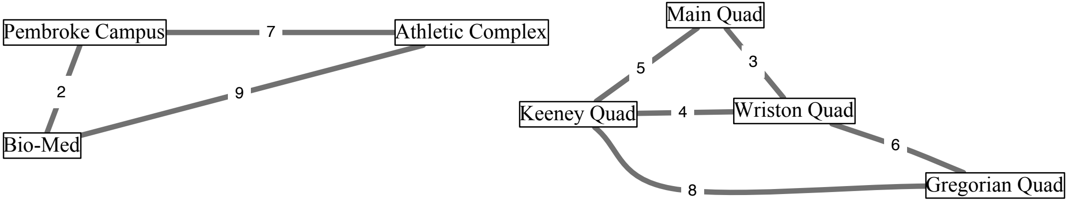
Minimum spanning forest problem:

- *input:* a graph G, and an assignment of real-number *weights* to the edges of G.
- *output:* a minimum-weight set S of edges that is spanning and a forest.

Application: Design hot-water delivery network for the university campus:

- Network must achieve same connectivity as input graph.
- An edge represents a possible pipe.
- Weight of edge is cost of installing the pipe.
- Goal: minimize total cost.

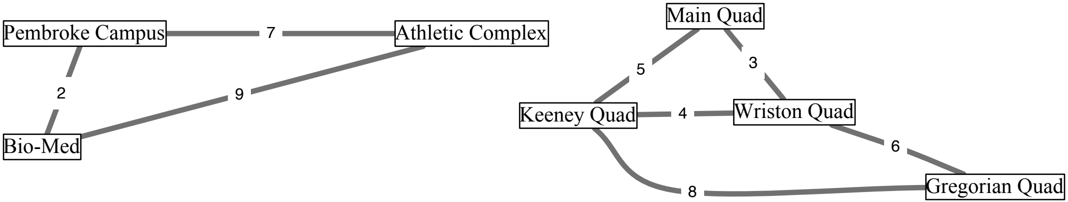
Minimum spanning forest: Grow algorithm



def GROW(G)
  $S := \emptyset$
  consider the edges in increasing order
  for each edge e:
    if e's endpoints are not yet connected
      add e to S.

Increasing order: 2, 3, 4, 5, 6, 7, 8, 9.

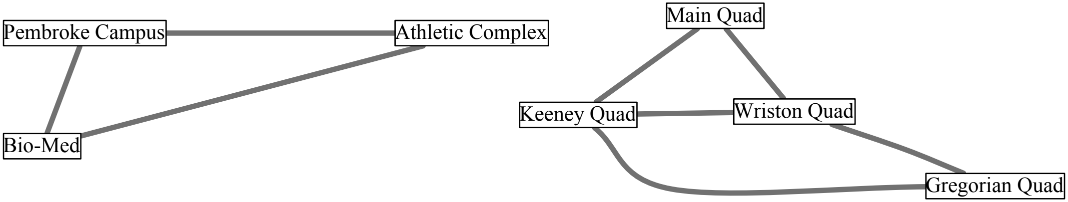
Minimum spanning forest: Shrink algorithm



def $\textsc{Shrink}(G)$
    $S = \{\text{all edges}\}$
    consider the edges in order, from highest-weight to lowest-weight
    for each edge e:
        if every pair of nodes are connected via $S - \{e\}$:
            remove e from S.

Decreasing order: 9, 8, 7, 6, 5, 4, 3, 2.

Formulating *Minimum Spanning Forest* in linear algebra

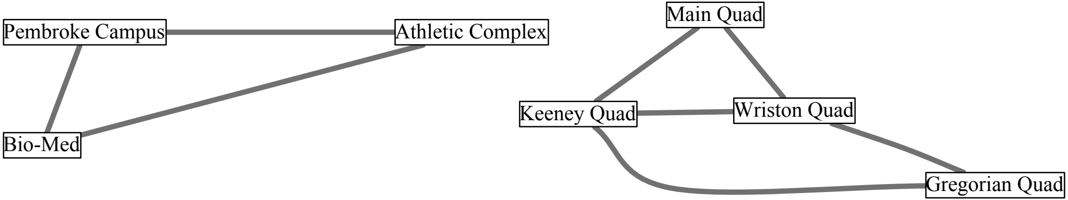


Let D = set of nodes {Pembroke, Athletic, Main, Keeney, Wriston}

Represent a subset of D by a $GF(2)$ vector:
subset `{Pembroke, Main, Gregorian}` is represented by

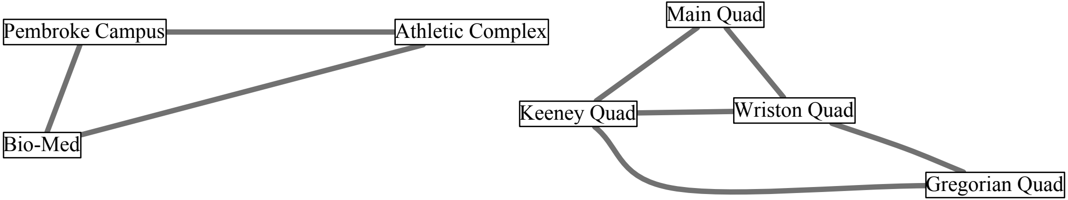
Pembroke	Athletic	Bio-Med	Main	Keeney	Wriston	Gregorian
1			1			1

Formulating *Minimum Spanning Forest* in linear algebra



edge	vector						
	Pembroke	Athletic	Bio-Med	Main	Keeney	Wriston	Gregorian
{Pembroke, Athletic}	1	1					
{Pembroke, Bio-Med}	1		1				
{Athletic, Bio-Med}		1	1				
{Main, Keeney}				1	1		
{Main, Wriston}				1		1	
{Keeney, Wriston}					1	1	
{Keeney, Gregorian}					1		1
{Wriston, Gregorian}						1	1

Formulating *Minimum Spanning Forest* in linear algebra



The vector representing {`Keeney`, `Gregorian`},

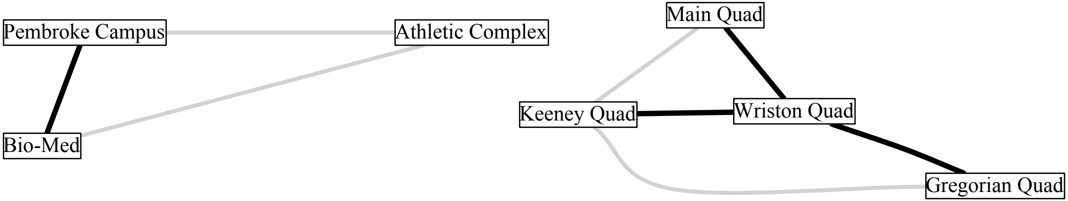
Pembroke	Athletic	Bio-Med	Main	Keeney	Wriston	Gregorian
				1		1

is the sum, for example, of the vectors representing {`Keeney`, `Main` }, {`Main`, `Wriston`}, and {`Wriston`, `Gregorian`} :

Pembroke	Athletic	Bio-Med	Main	Keeney	Wriston	Gregorian
			1	1		
			1		1	
					1	1

A vector with 1's in entries x and y is the sum of vectors corresponding to edges that form an x-to-y path in the graph.

Formulating *Minimum Spanning Forest* in linear algebra



A vector with 1's in entries x and y is the sum of vectors corresponding to edges that form an x-to-y path in the graph.

Example: The span of the vectors representing

{`Pembroke, Bio-Med`}, {`Main, Wriston`}, {`Keeney, Wriston`}, {`Wriston, Gregorian `}

- contains the vectors corresponding to

  {`Main, Keeney`}, {`Keeney, Gregorian`}, and {`Main, Gregorian`}

- but not the vectors corresponding to

  {`Athletic, Bio-Med `} or {`Bio-Med, Main`}.

Grow algorithms

def $\textsc{Grow}(G)$
    $S := \emptyset$
    consider the edges in increasing order
    for each edge e:
        if e's endpoints are not yet connected
            add e to S.

def $\textsc{Grow}(\mathcal{V})$
    $S = \emptyset$
    repeat while possible:
        find a vector $\mathbf{v}$ in $\mathcal{V}$ not in Span S,
        and put it in S.

- Considering edges e of G corresponds to considering vectors $\mathbf{v}$ in $\mathcal{V}$
- Testing if e's endpoints are not connected corresponds to testing if $\mathbf{v}$ is not in Span S.

The Grow algorithm for MSF is a specialization of the Grow algorithm for vectors.

Same for the Shrink algorithms.