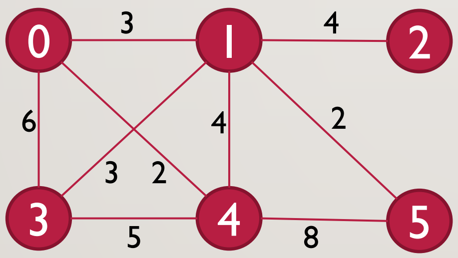
ANALYSIS OF ALGORITHMS

LECTURE 9 : MINIMUM WEIGHTED SPANNING TREES

BASED ON SECTION 5.2

WEIGHTED GRAPHS

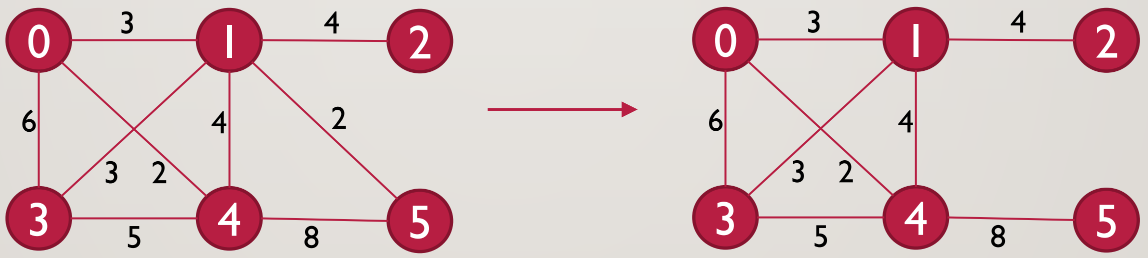
- This algorithm will consider weighted graphs
- In these graphs the edges have some weight attached.



- Meaning is problem specific

EXAMPLE : LOW COST COMPUTER NETWORK

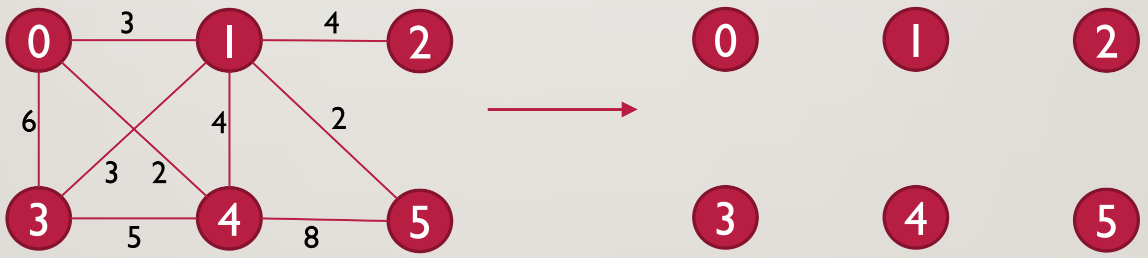
- We may want to construct a subgraph that uses less weight
- For example:



- That made it cost less but the vertices are still connected
- Could it still cost less though?

MINIMUM WEIGHTED GRAPHS

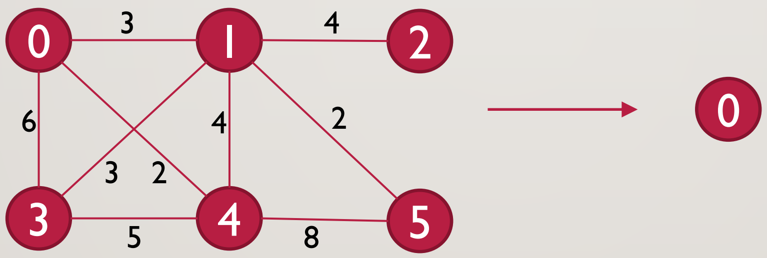
- What's the cheapest subgraph of our original graph?



- But we'd like the vertices to still be connected, otherwise it's kind of pointless
- Minimum Weighted Connected Graphs!

MINIMUM WEIGHTED CONNECTED GRAPHS

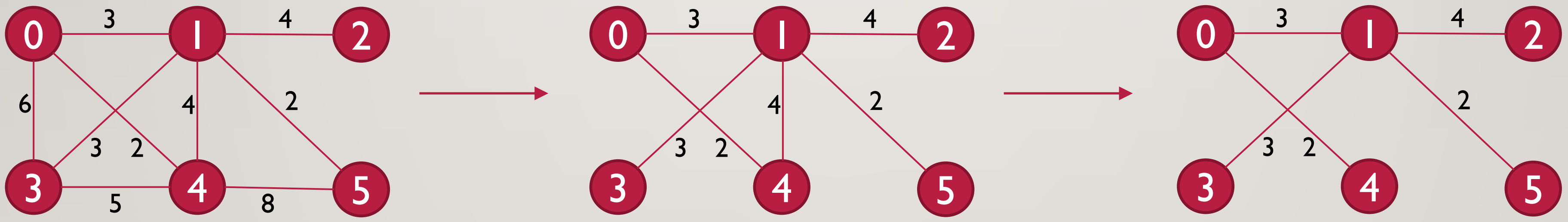
- What's the cheapest connected subgraph of our original graph?



- But we'd like to include all the original vertices, otherwise it's kind of pointless
- Minimum Weighted Connected Spanning Graphs

MINIMUM WEIGHTED CONNECTED SPANNING GRAPHS

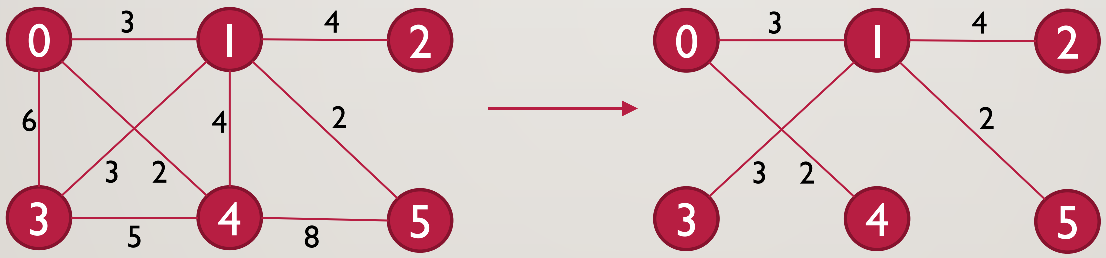
- What's the cheapest connected spanning subgraph of our original graph?
  - Is it possible to contain cycles?



- Minimum Weighted Connected Acyclic Spanning Graphs
- Minimum Weighted Spanning Trees

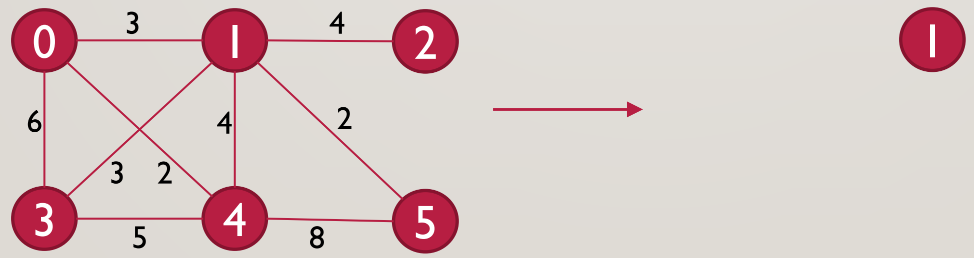
MINIMUM WEIGHTED SPANNING TREES

- How do we construct one?
- Greedy Algorithms



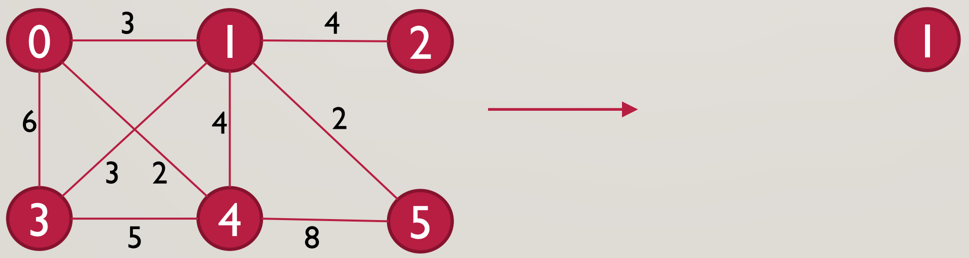
MINIMUM WEIGHTED SPANNING TREES

- How do we construct one?
- Start off with a tree with any one vertex



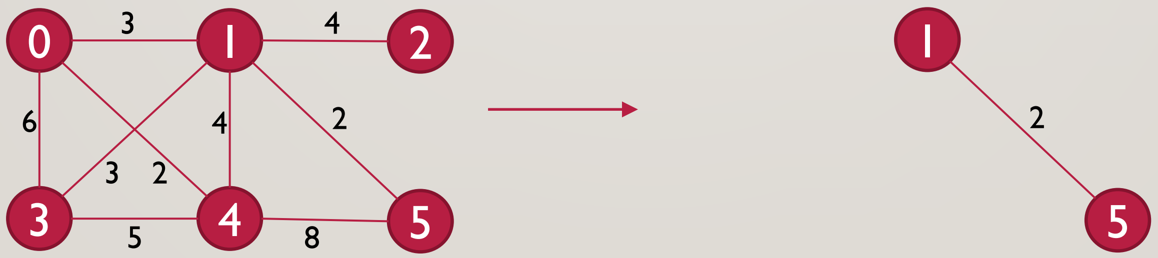
MINIMUM WEIGHTED SPANNING TREES

- How do we construct one?
- Start off with a tree with any one vertex
- Repeatedly
  - Add the cheapest edge from the original graph going from a vertex in the tree to one not in the tree
  - Stop when you have all the vertices from original tree



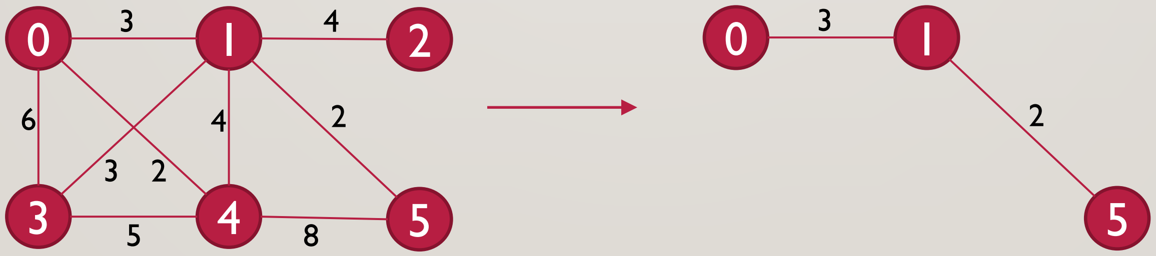
MINIMUM WEIGHTED SPANNING TREES

- How do we construct one?
- Start off with a tree with any one vertex
- Repeatedly
  - Add the cheapest edge from the original graph going from a vertex in the tree to one not in the tree
  - Stop when you have all the vertices from original tree



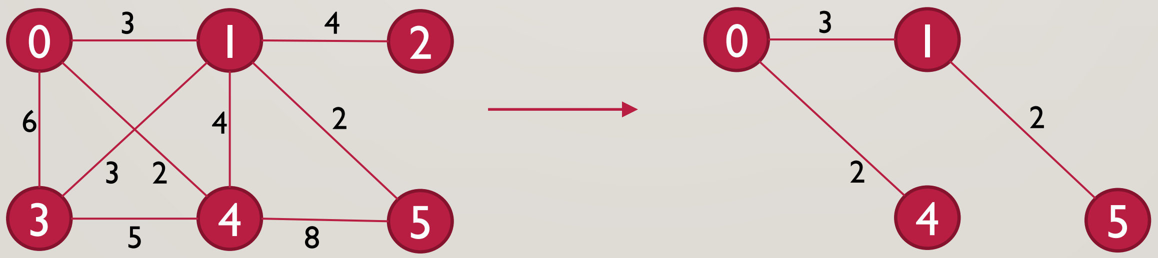
MINIMUM WEIGHTED SPANNING TREES

- How do we construct one?
- Start off with a tree with any one vertex
- Repeatedly
  - Add the cheapest edge from the original graph going from a vertex in the tree to one not in the tree
  - Stop when you have all the vertices from original tree



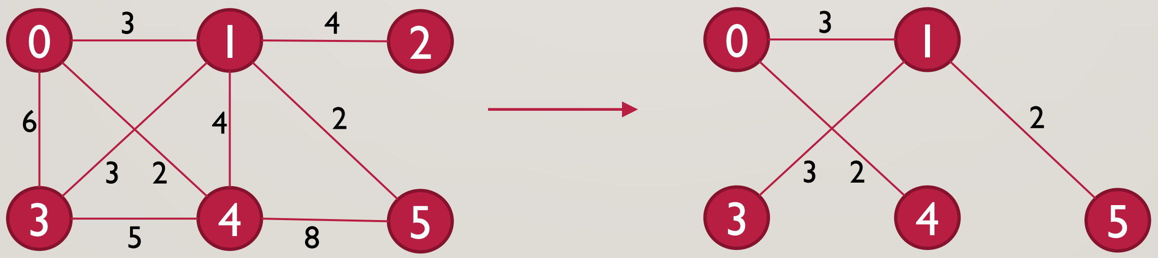
MINIMUM WEIGHTED SPANNING TREES

- How do we construct one?
- Start off with a tree with any one vertex
- Repeatedly
  - Add the cheapest edge from the original graph going from a vertex in the tree to one not in the tree
  - Stop when you have all the vertices from original tree



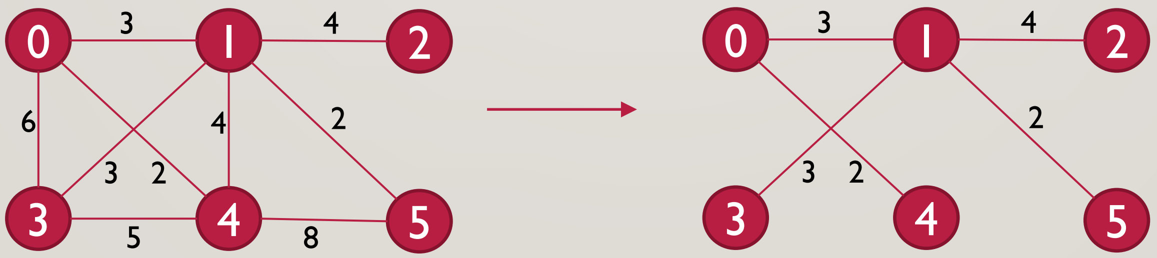
MINIMUM WEIGHTED SPANNING TREES

- How do we construct one?
- Start off with a tree with any one vertex
- Repeatedly
  - Add the cheapest edge from the original graph going from a vertex in the tree to one not in the tree
  - Stop when you have all the vertices from original tree



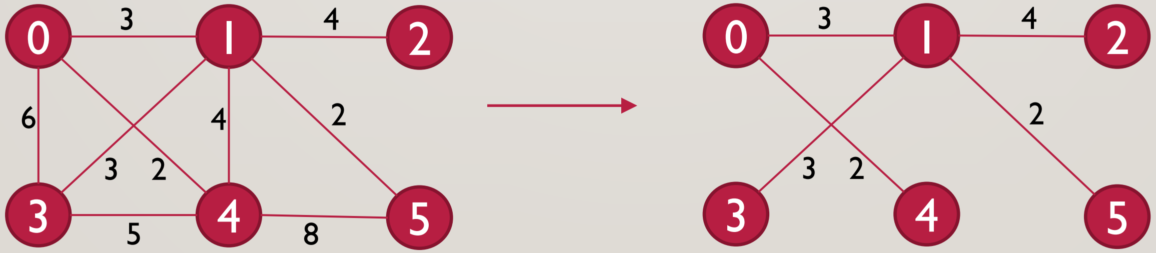
MINIMUM WEIGHTED SPANNING TREES

- How do we construct one?
- Start off with a tree with any one vertex
- Repeatedly
  - Add the cheapest edge from the original graph going from a vertex in the tree to one not in the tree
  - Stop when you have all the vertices from original tree



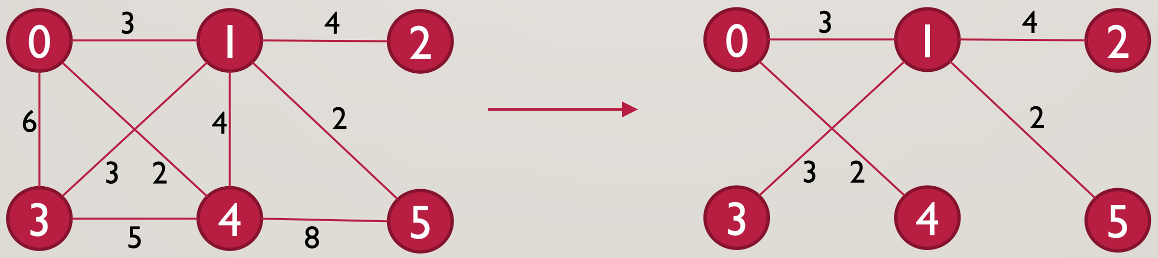
MINIMUM WEIGHTED SPANNING TREES

- How do we know this algorithm is correct?
    - Read and understand the proof in the book!



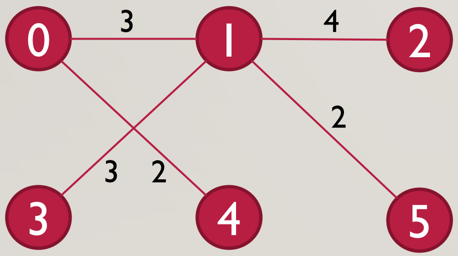
MINIMUM WEIGHTED SPANNING TREES

- How do we know this algorithm is correct?
  - Read and understand the proof in the book!
- But… How does that proof work?



MINIMUM WEIGHTED SPANNING TREES

- The intuition is that the algorithm is iterative, and after every iteration, we have a subtree of a minimum weighted spanning tree.
- So let's consider when we start. Our tree is a single vertex. Is this a subtree of the minimum weighted spanning tree?



MINIMUM WEIGHTED SPANNING TREES

- Now what we need is that if we start off with a subtree T' of a MWST and then we add the cheapest edge going from a vertex in T' to a vertex not in T', we end up with a new tree $T^{\#}$, which is also a subtree of a MWST
- This would be an important result, because what it's saying is that we can start of with a subtree of the MWST, and iteratively grow it.
- Since it is growing, it will eventually contain all the vertices, so it will be a MWST

MINIMUM WEIGHTED SPANNING TREES

- Now what we need is that if we start off with a subtree T' of a MWST and then we add the cheapest edge (x, y) going from a vertex in T' to a vertex not in T', we end up with a new tree $T^{\#} = T' + (x, y)$, which is also a subtree of a MWST
- Ok, so let's let T be a MWST of a graph G. Let T' be a subtree of that graph.
- Let the cheapest edge in G going from a vertex in T' to a vertex not in T' be (x, y)
- Ok, now there's two possibilities – Either (x, y) is in T or it's not in T
- If it is in T, then T' + (x, y) is a subtree of T, because T' was, and then we added something that was in T

MINIMUM WEIGHTED SPANNING TREES

- But what if (x, y) is NOT in T?
- Then T' $+ (x, y)$ is not a subtree of T.
- But we didn't say it had to be. We just said it had to be a subtree of a MWST. There could be many MWSTs of G
- Now, consider the path in T from x to y. It's not the edge (x, y) because (x, y) is not in T
- Let's write the path as $(x, v_1), (v_1, v_2), \dots, (v_n, y)$
- We know that x is in T' and y is not in T' because that's how we chose the edge
- So, on the path between x and y in T, there must be some edge that goes from a vertex in T' to a vertex not in T'. Call this edge (v_{i-1}, v_i)

MINIMUM WEIGHTED SPANNING TREES

- Now, construct a new tree T^*, which is $T - (v_{i-1}, v_i) + (x, y)$
- $T' + (x, y)$ is a subtree of T^*
- We want to show that T^* is another MWST of G
- First, is T^* a tree? Well, is it connected?
  - We removed (v_{i-1}, v_i). Can we still get from v_{i-1} to v_i?
  - Well, we know there was a path in T that went from x to v_{i-1} and from v_i to y
  - That means in T^* we can walk from v_{i-1} to x, then from x to y, then from y to v_i.
  - So its connected

MINIMUM WEIGHTED SPANNING TREES

- But is T^* a tree?
  - Well, T had $n - 1$ edges because it was a tree
  - Now we've removed an edge and added an edge
  - So T^* also has $n - 1$ edges
  - So it's connected and has $n - 1$ edges – That means it's a tree!

MINIMUM WEIGHTED SPANNING TREES

- Ok, but is T^* minimum weighted?
  - Well, T was minimum weighted
  - T^* = $T - (v_{i-1}, v_i) + (x, y)$
  - Now, we chose (x, y) as the cheapest edge that goes from a vertex in T' to a vertex not in T'
  - But (v_{i-1}, v_i) is an edge that goes from a vertex in T' to a vertex not in T'
  - So weight(x, y)<= weight (v_{i-1}, v_i)
  - That means that weight(T^*)<=weight(T)
  - But T was an MWST - "Minimum"
  - So weight(T^*)=weight(T)
  - So T^* is a MWST of G

MINIMUM WEIGHTED SPANNING TREES

- And $T' + (x, y)$ is a subtree of T^*
- So $T' + (x, y)$ is a subtree of a MWST