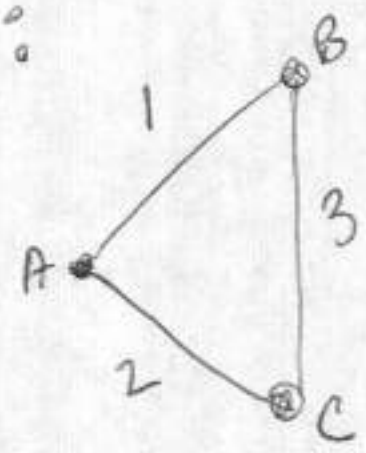
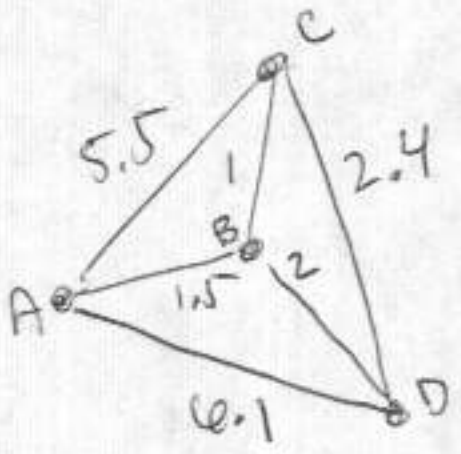
Ch.6 (cont'd): The Nearest Neighbor Algorithm

Definition: In a given weighted graph, a nearest neighbor of a given vertex is another vertex connected to the given vertex by an edge, such that the edge has the smallest possible weight.

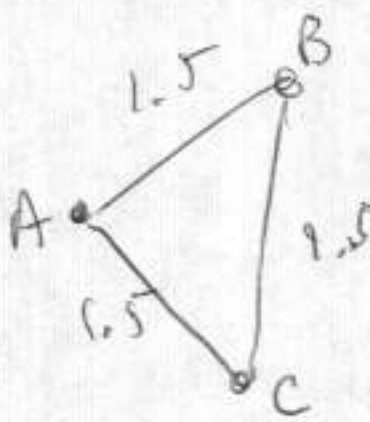
Example:



In this graph, a nearest neighbor for vertex A is vertex B. This is because the edge connecting A to B is of weight 1, smaller than the weight of the edge from A to C, which is 2.



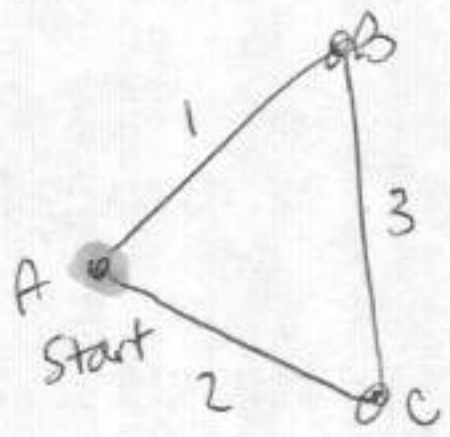
In this weighted graph, a nearest neighbor for B is the vertex C. A nearest neighbor for D is the vertex B.

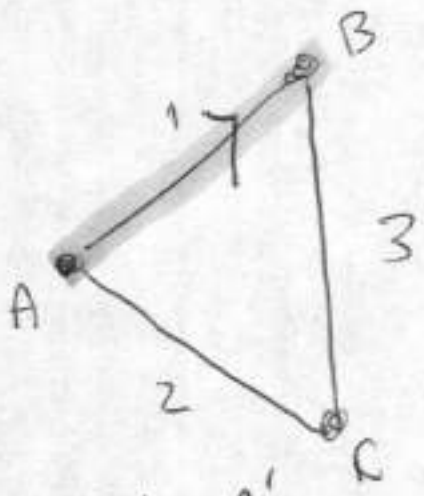


In this graph, both vertices B and C are nearest neighbors for A.

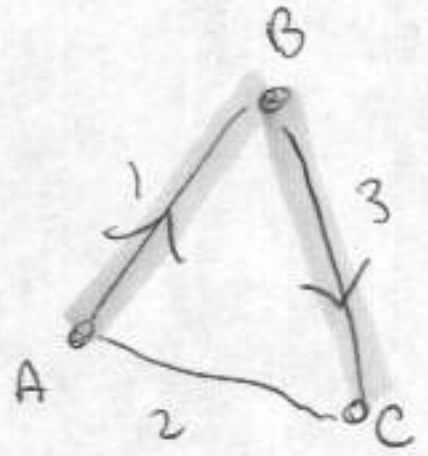
The Nearest Neighbor Algorithm is simple to desribe: with the goal of creating an optimal tour in our weighted graph, we begin at some starting vertex. Then we move to this vertex's nearest neighbor (or some nearest neighbor, if there's more than one). We continue in this fashion, until the Hamilton circuit is complete (at each step we move to the nearest neighbor which has not yet been travelled to).

We apply this to the above examples:

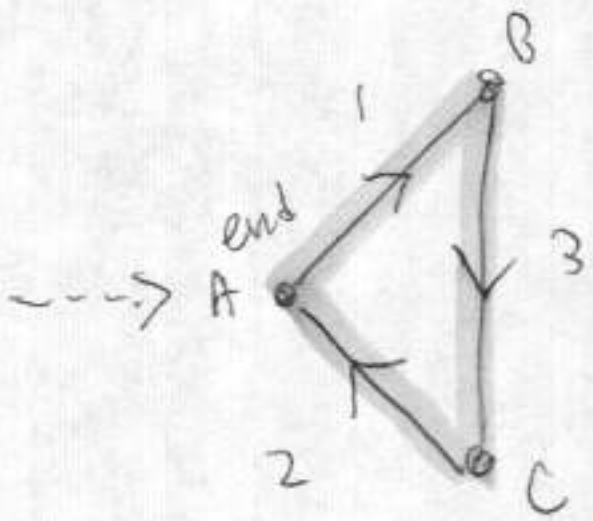




move to A's nearest neighbor

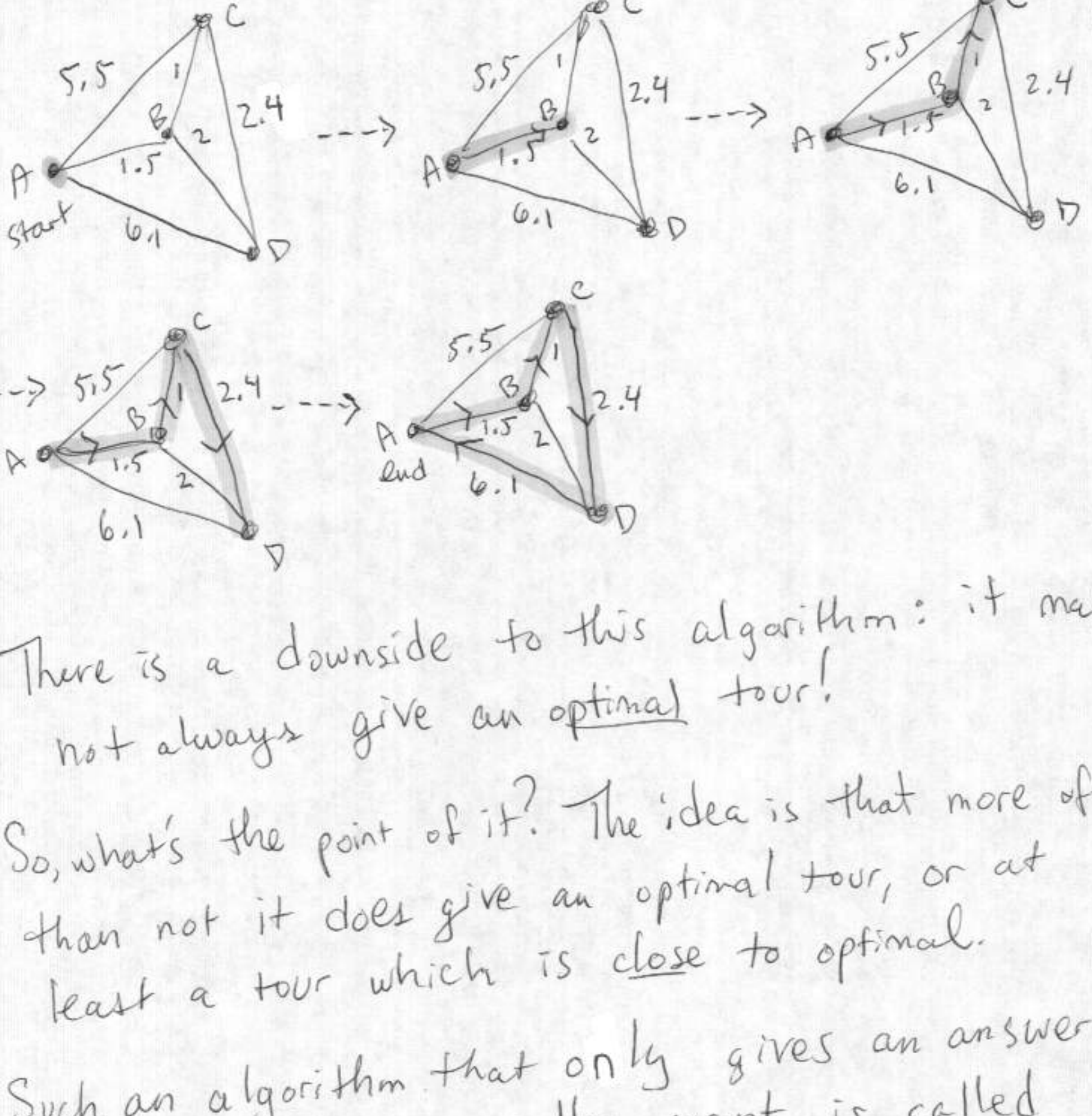


we have only one way of proceeding



only one choice again

This example is not very enlightening because the graph is so small.



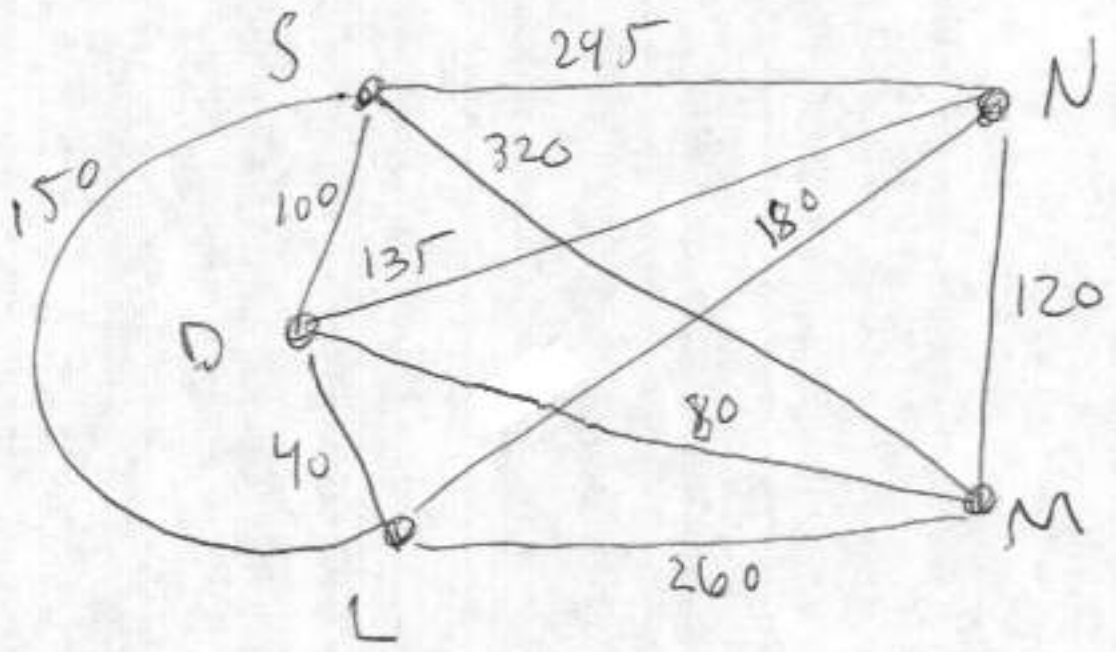
There is a downside to this algorithm: it may not always give an optimal tour!

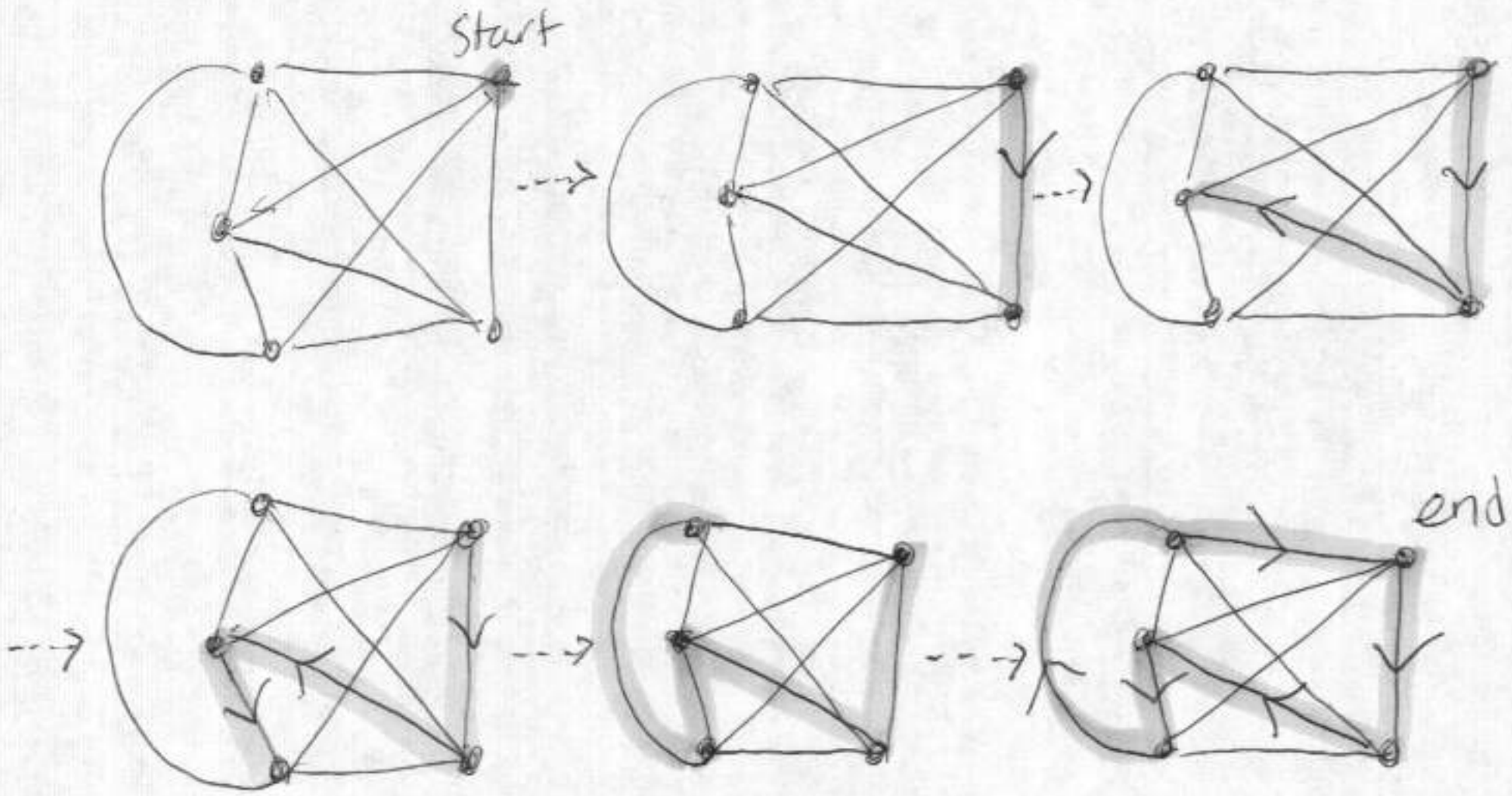
So, what's the point of it? The idea is that more often than not it does give an optimal tour, or at least a tour which is close to optimal.

Such an algorithm that only gives an answer close to what we really want is called an approximate algorithm.

This algorithm, and approximate algorithms in general are very useful when to get an actual solution requires much more computational work.

Let's apply the Nearest Neighbor Algorithm to our air travel problem from last time.





The cost of this tour is

$120 + $80 + $40 + $150 + $245 = $635.

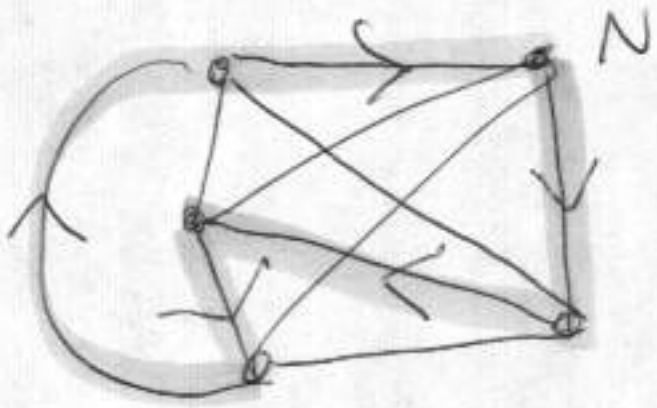
Recall that using the Brute Force Algorithm we found an optimal tour costing less, at $630.

But $635 is very close to optimal!

There is a variation of the Nearest Neighbor Algorithm: the <u>Repetitive Nearest Neighbor Algorithm.</u> (RNNA)

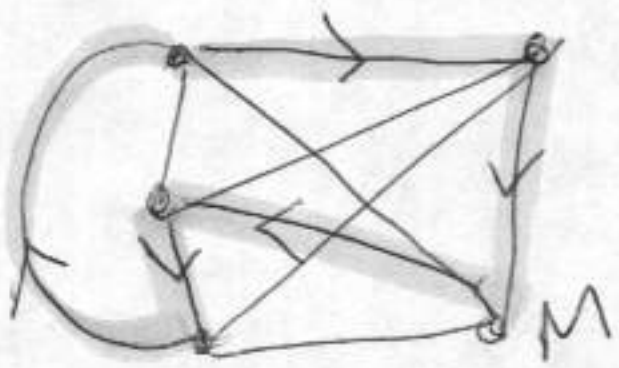
This is actually just doing the Nearest Neighbor Algorithm (NNA) more than once: the basic observation is that if we choose a different starting vertex, we may get a different tour using NNA! We then apply the NNA for each choice of different initial vertex, and then choose the best tour amongst those.

Let's apply the RNNA to our example:



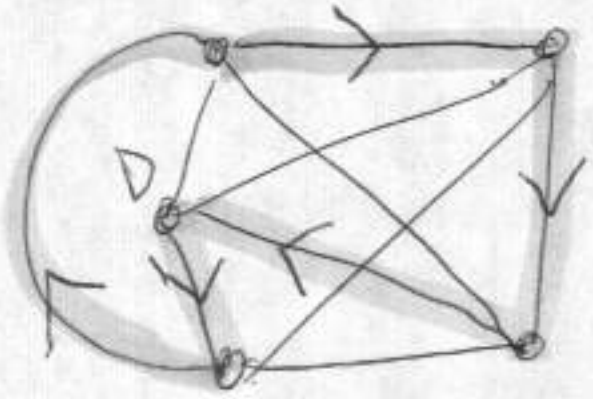
Start: NYC

NNA gives a tour at $635

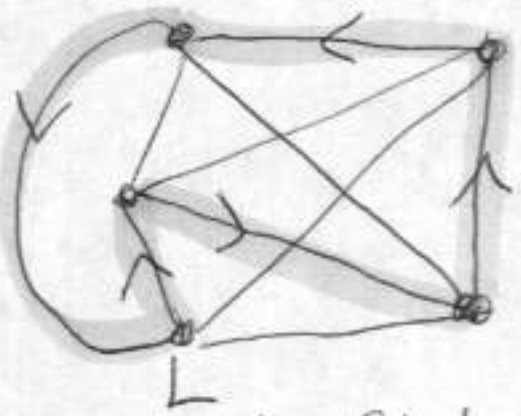


start: Miami

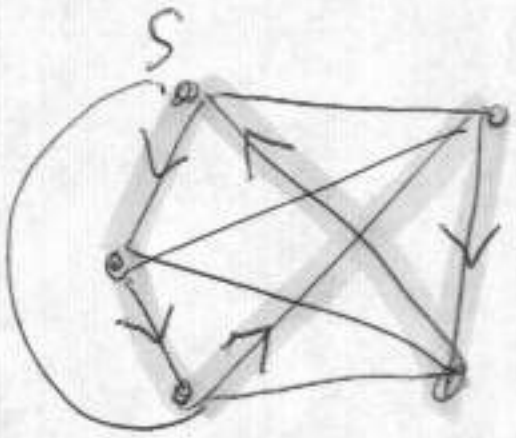
NNA gives a tour at $635



Start: Denver
NNA gives a tour
at $635



Start: St. Louis
NNA gives a tour
at $635



Start: Seattle
NNA gives a tour
at $760

They are all the same, except the last, which is certainly not optimal.

Thus the RNNA may still not give an optimal tour, but performs better (in general) than just one implementation of the NNA (although in our example it happened to not improve our answer).