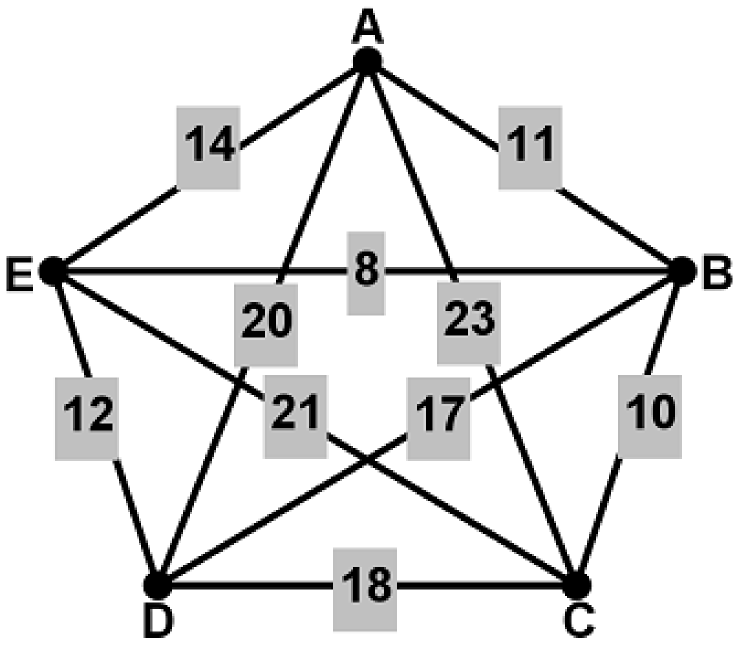
Here is a tip for sorted edges:

At the beginning, you may want to make a list of the vertices, and a list of the edge weights. Then sort the list of edge weights from the smallest number to the largest. You may also want to write down which vertices are at the end of each edge. For example, here is such a list for the graph from class:



Vertices:

A, B, C, D, E

Edge weights:

8 {B, E}

10 {B, C}

11 {A, B}

12 {D, E}

14 {A, E}

17 {B, D}

18 {C, D}

20 {A, D}

21 {C, E}

23 {A, C}

As you pick edges, you can mark your vertices on a copy of the graph or on the list of vertices:

1) Put a single slash (\ ) through a vertex if it is connected to ONE edge you have picked.

2) Cross out (X out) a vertex if it is connected to TWO edges you have picked.

As you look around the entire graph for the cheapest available edge, the vertex marks can help you avoid problems:

1) Do not pick edges connected to vertices you have crossed out. This prevents picking three edges at the same vertex.

Cross out edges that are connected to vertices you have crossed out.

2) Be suspicious if the next cheapest edge you might want to pick has slashed vertices at each end. What you can do is temporarily pick this edge, then use your finger to try to trace a path along the edges you have picked so far. If you are able to start at a vertex, trace your finger along the edges you have picked, and return to that vertex, your path goes in a circle (loop), which means that the edge you just (temporarily) picked was bad. You must skip it and try the next cheapest edge. If your path does not go in a circle, the edge you just (temporarily) picked is good, and should be kept.

If you skip an edge, cross it out on your list so you do not accidently pick it later.