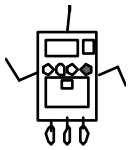
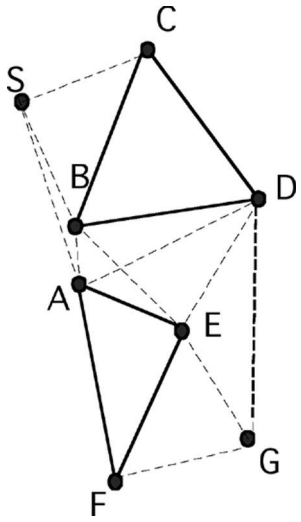
6.034 Recitation 4: Another Optimal Search Exercise (9/25/03)

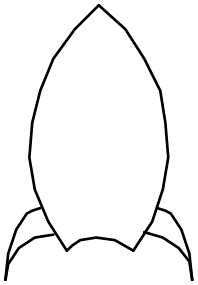
LOrtiz (Orig. by KKoile)

A* to the rescue!

Wallace and Gromit have just finished their vacation on the moon and are about to head back to Earth in their rocket ship (located at position G below). The local robot desperately wants to go back with them, but must hurry to get to the rocket ship in time. (He's at S below.) He has to navigate around two obstacles (shown as triangles AEF and BCD.) He uses his nifty A* search engine to find the best path. Which way does he go?







Link lengths:			
S-A	6	B-D	6
S-B	5	B-E	5
S-C	4	C-D	6
A-B	1	D-E	5
A-D	7	D-G	8
A-E	3	E-F	6
A-F	7	E-G	4
B-C	6	F-G	4

Estimates of distance to G from:	
A	7
B	9
C	13
D	7
E	4
F	4
G	0
S	1

1. Assume his A* algorithm uses an extended list, adds new elements to the front of the queue, and breaks ties in choice of node to extend by picking the one closer to the front of the queue. Assume all heuristics are admissible (even if they aren't). What's the order of node extension, and what's the path? (And if you've seen the animation short, does he make it to the rocket ship in time?)

Footnote: This sort of graph is called a visibility graph and is commonly used for obstacle avoidance in robot navigation problems. The idea is this: Figure out how close the robot could get to each vertex of each obstacle, make those new locations nodes in your search space (called a configuration space), connect each node to all other ones that are visible from it (i.e. add links between the nodes visible from one another), then search the resulting graph for a path, or the shortest path, from a start location to an end location.