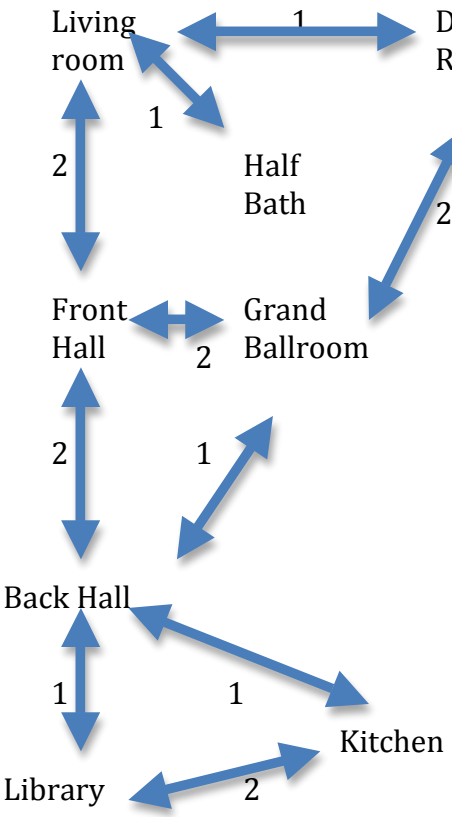
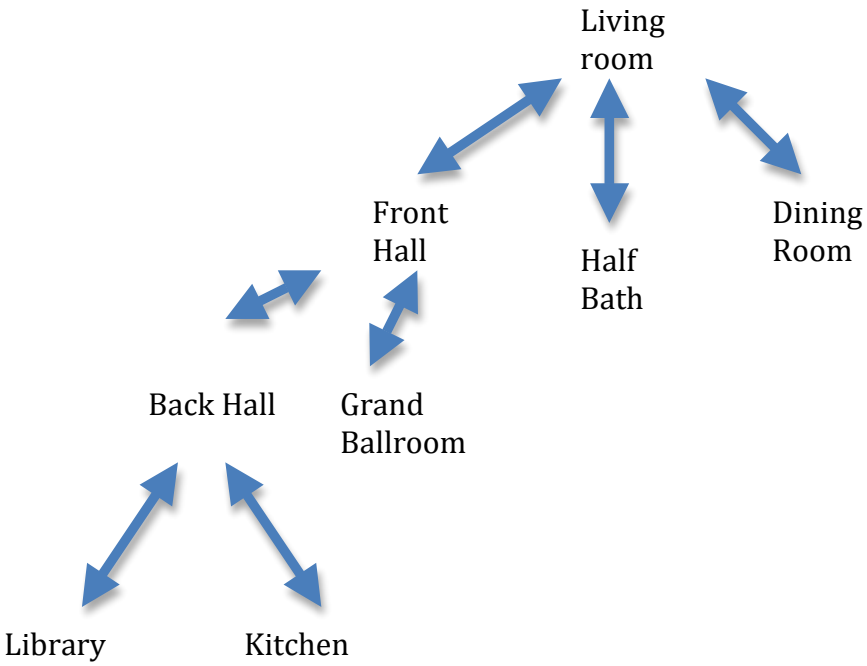
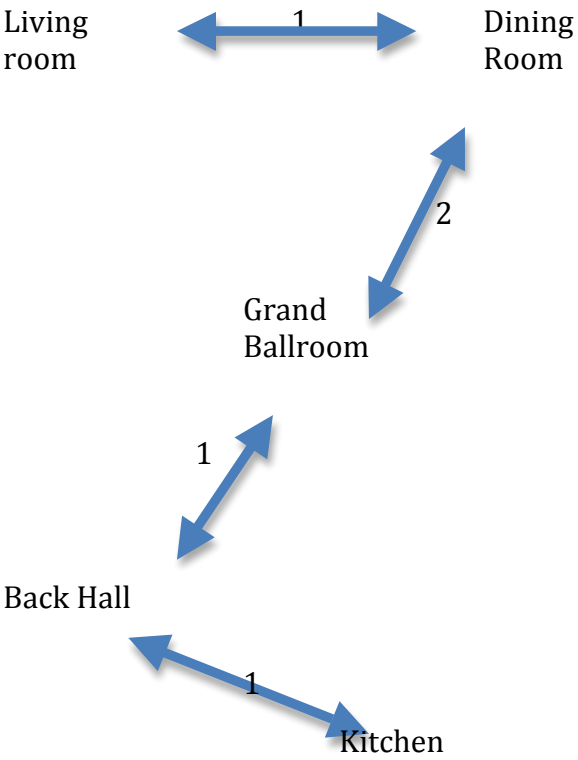
1.



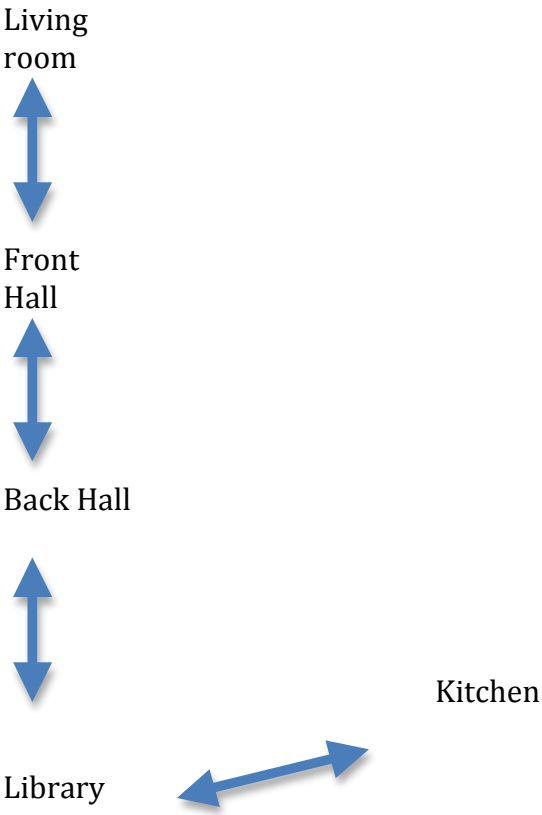
2.Breadth first search:



3.


Living room
1
Dining Room
2
Grand Ballroom
1
Back Hall
1
Kitchen

4.


Living room
Front Hall
Back Hall
Kitchen
Library

5.

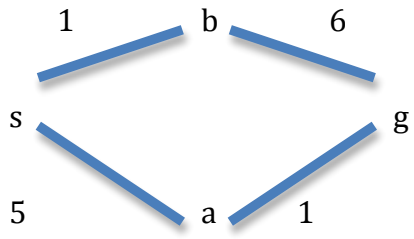
	DFS	Nodes	Time(nano)	Cost	BFS	Nodes	Time(nano)	Cost	A*	Nodes	Time(nano)	Cos
Map 1		11	140000	7		7	1020000	7		8	138000	7
Map 2		25	181000	14		14	152000	14		15	210000	14
Map 3		33	213000	14		33	252000	14		28	325000	14
Map 4		27	247000	14		23	151000	14		24	224000	14
Map 5		55	244000	14		45	318000	14		46	581000	14
Map 6		40266	49308000	10092		979801	251407000	10090		973516	459370000	10090
Map 7	NO Path!				NO Path!				NO Path!			
Map 8		25	192000	13		23	193000	13		16	223000	13

6. I first start with the goal node, it maintains a priority queue of nodes to be traversed, the open set. The lower fcost is, the higher its priority. At each step, the node with lowest fcost is removed from priority queue. The algorithm continues until the node is found. My method is admissible because it never overestimates the cost of reaching the goal. I used the Euclidean distance, which is the distance between the current node and goal node. It will always be less the actual cost because first we can only move up, down, left or right, we can not move diagonally. It is very useful especially for the later ones because there are obstacles in the way. A* is very helpful if there are many obstacles between the start and the goal.

7.The way question is worded is a little confusing. I interpreted it as when DFS is more efficient than BFS. When the goal is many hops away from the start, DFS would succeed but BFS would fail. Because it would take BFS to basically look through all the nodes in the graph to find goal, on the other hand, DFS may find it right away.

8. The way question is worded is a little confusing. I interpreted it as when BFS is more efficient than DFS. When the goal is next to the start node, BFS would succeed but DFS would fail. Because DFS may basically look through all the nodes in the graph to find the goal, because it keeps expanding the nodes along a path, on the other hand, BFS first take look at start node's neighbors.

9. Lets say we have four nodes, s,a,b,,g . A* may choose s to b to g. But BFS chooses s to a to g, which costs less.



10.
Map1: The heuristic is admissible. It is not very useful because every node basically has only one neighbor(not counting the one we already visited). So we have to choice but to keep going down the only path.
Map2: The heuristic is admissible. It is not very useful because every node basically has only one neighbor(not counting the one we already visited). So we have to choice but to keep going down the only path.
Map3: The heuristic is admissible. It is pretty useful because there are different paths to the goal node and my A* is able to figure out which way will get me there fastest. I only explored 28 nodes to get to the goal.
Map4: The heuristic is admissible. It is pretty useful compared to DFS, because there are different paths to the goal node and my A* is able to figure out which way will get me there fastest. I only explored 24 nodes to get to the goal compared 24 nodes by DFS.
Map5: The heuristic is admissible. It is pretty useful compared to DFS, because there are different paths to the goal node and my A* is able to figure out which way will get me there fastest. I only explored 46 nodes to get to the goal compared 45 nodes by DFS. BFS and A* are about the same.

Map6: The heuristic is admissible. It is not very useful compared to DFS, mainly because there are no obstacles from the start to the goal. A* will not help us a lot in this map. However, the greedy search will help us a lot in this case.
Map7: The heuristic is admissible. But since there is no actual path from start to the goal, I can not say if A* is helpful or not.
Map8: The heuristic is admissible. It is very useful compared to both DFS and BFS. It only took A* 16 nodes to find the goal, but took 25 and 23 for DFS and BFS respectively.
In general, A* is helpful to us but in some extreme cases, BFS and DFS are better.