Ryan Smith
CS 540: AI
HW #2

2.

(a)
Breadth-First search

Step#	OPEN	CLOSED	X	CHILDREN	RemainingCHILDREN
1	{S}	{}	S	{A, C}	{A, C}
2	{A, C}	{S}	A	{B, G1}	{B, G1}
3	{C, B, G1}	{S, A}	C	{D, F, J}	{D, F, J}
4	{B, G1, D, F, J}	{S, A, C}	B	{G1, J}	{}
5	{G1, D, F, J}	{S, A, C, B}	G1	{}	{}

Goal state reached: G1
States popped off of the OPEN list: S, A, C, B, G1

(b)
Depth-First search

Step#	OPEN	CLOSED	X	CHILDREN	RemainingCHILDREN
1	{S}	{}	S	{A, C}	{A, C}
2	{A, C}	{S}	A	{B, G1}	{B, G1}
3	{B, G1, C}	{S, A}	B	{G1, J}	{J}
3	{J, G1, C}	{S, A, B}	J	{G1, F}	{F}
4	{F, G1, C}	{S, A, B, J}	F	{G2}	{G2}
5	{G2, G1, C}	{S, A, B, J, F}	G2	{}	{}

Goal state reached: G2
States popped off of the OPEN list: S, A, B, J, F, G2

(c)
Iterative Deepening (depth limit 0)

Step#	OPEN	CLOSED	X	CHILDREN	RemainingCHILDREN
1	{S}	{}	S	{A, C}	{A, C}

Iterative Deepening (depth limit 1)

Step#	OPEN	CLOSED	X	CHILDREN	RemainingCHILDREN
1	{S}	{}	S	{A, C}	{A, C}
2	{A, C}	{S}	A	{B, G1}	{B, G1}
3	{C, B, G1}	{S, A}	C	{D, F, J}	{D, F, J}

Iterative Deepening (depth limit 2)

Step#	OPEN	CLOSED	X	CHILDREN	RemainingCHILDREN
1	{S}	{}	S	{A, C}	{A, C}

```
2       {A, C}          {S}             A       {B, G1}         {B, G1}
3       {B, G1, C}      {S, A}          B       {G1, J}         {J}
4       {G1, C, J}      {S, A, C}       G1      {}              {}

Goal state reached: G1
States popped off of the OPEN list (depth limit 0): S
States popped off of the OPEN list (depth limit 1): S, A, C
States popped off of the OPEN list (depth limit 2): S, A, B, G1

(d)
Uniform Cost (i.e., using f = g)
Step#   OPEN                            CLOSED                                  X       CHILDREN                RemainingCHILDREN
1       {S_0}                           {}                                      S       {A_4, C_3}              {A_4, C_3}
2       {C_3, A_4}                      {S_0}                                   C       {D_5, F_10, J_12}       {D_5, F_10,
J_12}
3       {A_4, D_5, F_10, J_12}          {S_0, C_3}                              A       {B_7, G1_12}            {B_7, G1_12}
4       {D_5, B_7, F_10, G1_12, J_12}   {S_0, C_3, A_4}                         D       {E_7, F_6}              {E_7, F_6}
5       {F_6, B_7, E_7, J_12, G1_12}    {S_0, C_3, A_4, D_5}                    F       {G2_8}                  {G2_8}
6       {B_7, E_7, G2_8, J_12, G1_12}   {S_0, C_3, A_4, D_5, F_6}               B       {G1_11, J_9}            {G1_11, J_9}
7       {E_7, G2_8, J_9, G1_11}         {S_0, C_3, A_4, D_5, F_6, B_7}          E       {F_10, G2_13}           {}
8       {G2_8, J_9, G1_11}              {S_0, C_3, A_4, D_5, F_6, B_7, E_7}     G2      {}
{}

Goal state reached: G2
States popped off of the OPEN list: S, C, A, D, F, B, E, G2

(e)
Best-First (using f = h)
Step#   OPEN                            CLOSED                  X       CHILDREN                RemainingCHILDREN
1       {S_6}                           {}                      S       {A_8, C_2}              {A_8, C_2}
2       {C_2, A_8}                      {S_6}                   C       {D_5, F_2, J_1}         {D_5, F_2, J_1}
3       {J_1, F_2, D_5, A_8}            {S_6, C_2}              J       {G1_0, F_2}             {G1_0}
4       {G1_0, F_2, D_5, A_8}           {S_6, C_2, J_1}         G1      {}                      {}

Goal state reached: G1
States popped off of the OPEN list: S, C, J, G1

(f)
Best-First (usiing f = g + h)
Step#   OPEN                                    CLOSED                                  X       CHILDREN
RemainingCHILDREN
1       {S_(0+6)}                               {}                                      S       {A_(4+8), C_(3+2)}
{A_(4+8), C_(3+2)}
2       {C_(3+2), A_(4+8)}                      {S_(0+6)}                               C       {D_(5+5), F_(10+2),
J_(12+1)}       {D_(5+5), F_(10+2), J_(12+1)}
```

3	{D_(5+5), A_(4+8), F_(10+2), J_(12+1)}	{S_(0+6), C_(3+2)}	D	{E_(7+3), F_(6+2)}	{E_(7+3), F_(6+2))}
4	{F_(6+2), E_(7+3), A_(4+8), J_(12+1)}}	{S_(0+6), C_(3+2), D_(5+5)}	F	{G2_(8+0)}	{G2_(8+0)}
5	{G2_(8+0), E_(7+3), A_(4+8), J_(12+1)}	{S_(0+6), C_(3+2), D_(5+5), F_(6+2)}	G2	{}	{}

Goal state reached: G2
States popped off of the OPEN list: S, C, D, F, G2

(g)
Beam Search (with beam width = 2 and f = h)

Step#	OPEN	CLOSED	X	CHILDREN	RemainingCHILDREN
1	{S_0}	{}	S	{A_8, C_2}	{A_8, C_2}
2	{C_2, A_8}	{S_0}	C	{D_5, F_2, J_1}	{D_5, F_2, J_1}
3	{J_1, F_2}	{S_0, C_2}	J	{F_2, G1_0}	{G1_0}
4	{G1_0, F_2}	{S_0, C_2, J_1}	G1	{}	{}

Goal state reached: G1
States popped off of the OPEN list: S, C, J, G1

(h)
Hill Climbing (using the h function only)

Step#	OPEN	CLOSED	X	CHILDREN	RemainingCHILDREN
1	{S_6}	{}	S	{A_8, C_2}	{A_8, C_2}
2	{C_2}	{S_6}	C	{D_5, F_2, J_1}	{D_5, F_2, J_1}
3	{J_1}	{S_6, C_2}	J	{F_2, G1_0}	{F_2, G1_0}
4	{G1_0}	{S_6, C_2, J_1}	G1	{}	{}

Goal state reached: G1
States popped off of the OPEN list: S, C, J, G1

(i)
No, this h function is not admissible. The h value at D is 5, and there is a path from D to a goal node with cost 1 + 2 = 3 (D -> F -> G2). This cost is less than the h value at D, meaning that h overestimates the cost to get from D to a goal node. Therefore, h is not admissible.

3.

(a)
If you are at Node C and simulated annealing has randomly selected node J for consideration, what is the probability this node is accepted?

Using simulated annealing we would actually go straight to J without needing to calculate a probability since J's score is better than C's. The probability of going to J is therefore 1. For good measure, below is what would be the probabilitiy calculation, which comes out to a value greater than 1 (which doesn't make sense as a probabiliity).

$$e^{((score(J) - score(C)) / Temperature)} = e^{(((-1) - (-2)) / 100)}$$

$$= e^{(1/100)}$$

$$\approx 1.01005016708$$

(b)
If you are at Node C and simulated annealing has randomly selected node D for consideration, what is the probability this node is accepted?

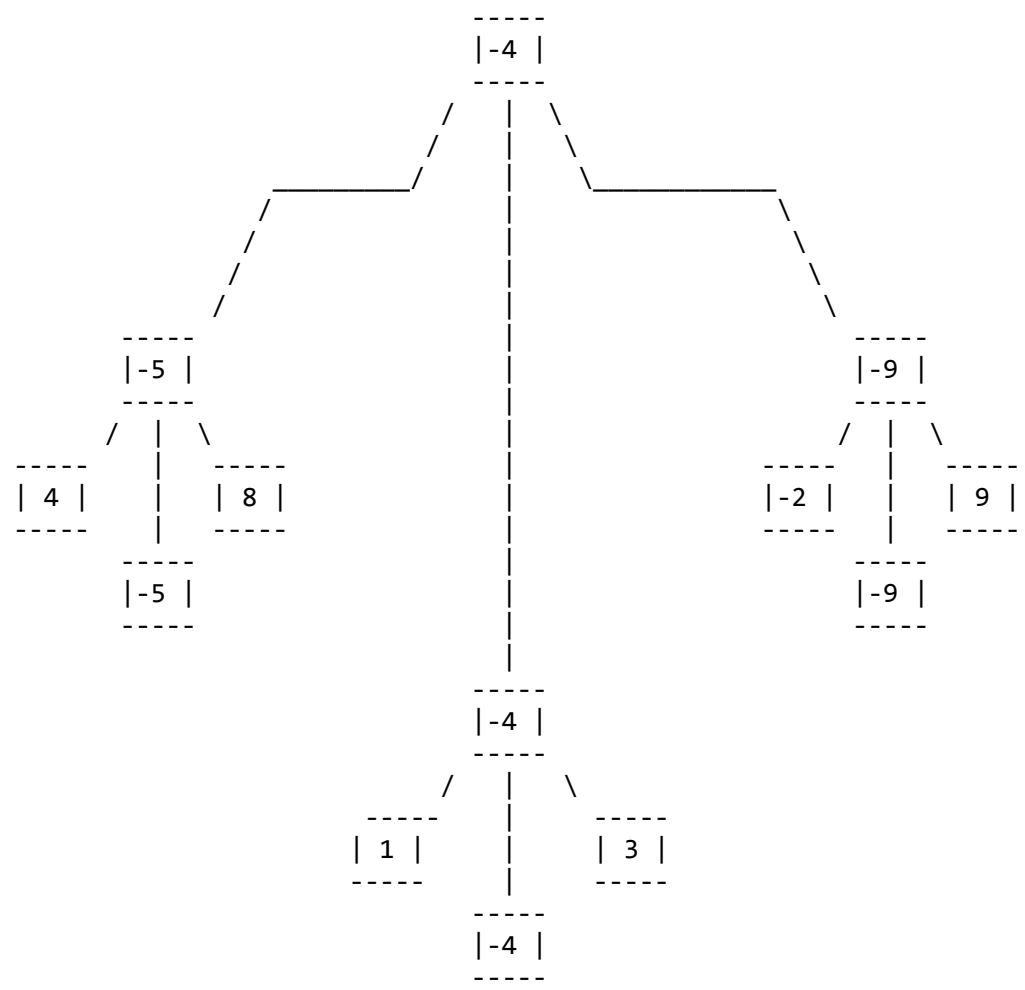
$$e^{((score(D) - score(C)) / Temperature)} = e^{(((-5) - (-2)) / 100)}$$

$$= e^{(-(3/100))}$$

$$\approx 0.97044553354$$

4.

(a)



Rather than circle it I'll just call it out - the ccorrect move for the maximizer would be to pick the middle arc, going to the intermediate node where I've put -4.

(b)
List one (1) leaf node in the above game tree whose SBE score need not be computed. Explain why.

One such node is the leaf node containing a positive 9.

While it would be great for the maximizer to achieve this value, once we evaluate the leaf noode with a value of -9 that shares a parent, we know it won't be possible. If we label the node with value 9 X - the minimizer wouldd chose either the -9

node or X if it turns out X's value is less than -9. Either way, the node in the parent that these twoo share comes out to at most -9. With this information, the maximizer wouldn't chose the parent no matter the value of X, so there's no need to calculate this node's value.