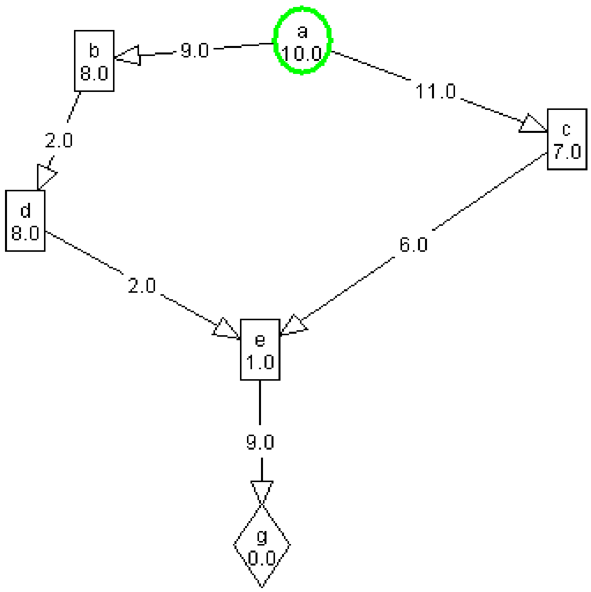
Question 2 [15 marks]

Consider the following graph, with arcs costs shown on the arcs and values of the heuristic function *h* shown inside the nodes. We want to find the optimal path from a to G using A*



1. For each iteration of A*, show the node removed from the frontier and the nodes added to the frontier along with the paths from a to these nodes. Give the g-value, h-value and f-value for each node added to the frontier. Please continue the trace provided and show clearly the path found. [7 marks]

TRACE

Number	Path	g	h	f
1	a	0	10	| 10

Remove path number 1

2. a -> b ....9..............8...........17................

3. a -> c ....11............7...........18..........................

Remove path number .....2...

4......................a -> b -> d..............11...........8............19...

Remove path number ..3..

5......................a -> c -> e.............17............1...........18...

Remove path number ...5...

6 ....a -> c -> e -> g........26............0............26..

Remove path number ....4...

7 ....a -> b -> d -> e........13...........1............14...

Remove path number ....7...

8 ....a -> b -> d -> e -> g.........22...........0.............22...

Remove path number .....8...

.... Reached Goal..

............Final path: a -> b -> d -> e -> g...

....

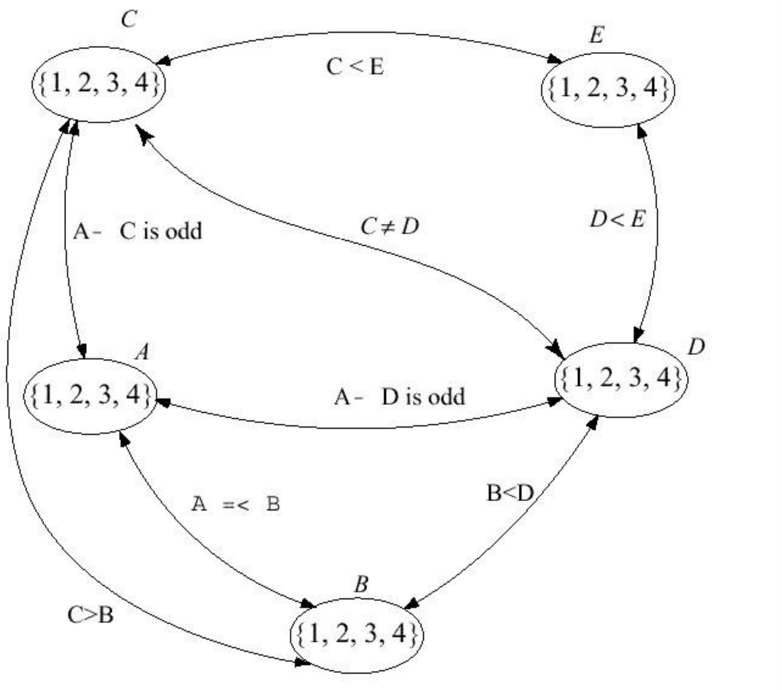
2. Give the path that Best First Search finds on the above graph. Is the path optimal in terms of arc cost? Explain your result [3 marks]

Path: a -> c -> e -> g

Path is non-optimal, and Best First Search finds it because it relies only on h(n) to decide which nodes to expand. Thus, it gets misled by the fact that the heuristic value of c is lower than that of b, even though the arc cost of the path from c to g is higher than the arc cost of the path from b to g.

Question 3 [12 marks]

Consider a constraint satisfaction problem (CSP) with five variables, A , B , C , D , and E , each with domain {1, 2, 3, 4}. We have eight constraints: A ≤ B; A - C is odd; C >B; A-D is odd; B <D; C ≠D; C <E; D <E . This CSP is depicted in the following constraint network:



a) [6 marks] On the constraint network above, cross out those values for each variable that are removed by arc consistency. In the space below, indicate which arc has been used to delete each value. For example, if you have the arc labeled (X<Y) and this causes the value *Val1* to be removed from the domain of variable X, you write: **Arc <X, Y> removes *Val1* from X**. If arc (Y>X), is used to remove *Val2* from the domain of variable Y, you write **Arc <Y, X> removes *Val2* from Y.** Do not do any domain splitting.

arc <C,(C,E)>	removes 4	from C
arc <E,(C,E)>	removes 1	from E
arc <D,(D,E)>	removes 4	from D
arc <D,(D,B)>	removes 1	from D
arc <B,(D,B)>	removes 3,4	from B
arc <C,(C,B)>	removes 1	from C
arc <A,(A,B)>	removes 3,4	from A
arc <E,(D,E)>	removes 2	from E

Domains become: A = {1,2}, B = {1,2}, C={2,3}, D={2,3}, E={3,4}

b) [4 marks] Does this CSP have a solution? Either give a solution or explain why it does not have one.

No solution, because the constraints C != D, A-C is odd, and A-D is odd can't be satisfied simultaneously with the domain values left (A={1,2}, C={2,3}, D={2,3})