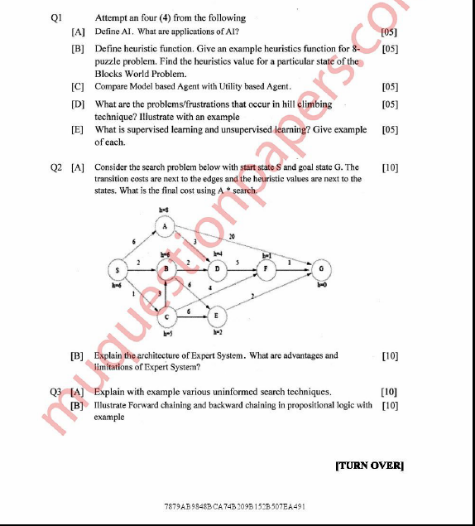
**Q.2 (a) Consider the search problem below with start state S and goal state G. The transition costs are next to the edges and the heuristic values are next to the states. What is the final cost using A\* search.**



At initialization it got three paths,

S -> A = g(n) + h(n) = 6 + 8 = 14

S -> B = g(n) + h(n) = 2 + 6 = 8

S -> C = g(n) + h(n) = 1 + 5 = 6

1st Iteration:

S -> A -> G = g(n) + h(n) = 6 + 20 + 0 = 26 (goal state reached)

S -> A -> D = g(n) + h(n) = 6 + 3 + 4 = 13

S -> B -> D = g(n) + h(n) = 2 + 2 + 4 = 8

S -> B -> E = g(n) + h(n) = 2 + 6 + 2 = 10

S -> C -> E = g(n) + h(n) = 1 + 6 + 2 = 9

S -> C -> B = g(n) + h(n) = 1 + 3 + 6 = 10

S -> C -> F = g(n) + h(n) = 1 + 4 + 1 = 6

2nd Iteration:

S -> A -> D -> F = g(n) + h(n) = 6 + 3 + 5 + 1 = 15

S -> B -> D -> F = g(n) + h(n) = 2 + 2 + 5 + 1 = 10

S -> B -> E -> G = g(n) + h(n) = 2 + 6 + 2 + 0 = 10 (goal state reached)

S -> C -> E -> G = g(n) + h(n) = 1 + 6 + 2 + 0 = 9 (goal state reached)

S -> C -> F -> G = g(n) + h(n) = 1 + 4 + 1 + 0 = 6 (goal state reached)

S -> C -> B -> D = g(n) + h(n) = 1 + 3 + 2 + 4 = 10

3rd Iteration:

S -> A -> D -> F -> G = g(n) + h(n) = 6 + 3 + 5 + 1 + 0 = 15 (goal state reached)

S -> B -> D -> F -> G = g(n) + h(n) = 2 + 2 + 5 + 1 + 0 = 10 (goal state reached)

S -> C -> B -> D -> F = g(n) + h(n) = 1 + 3 + 2 + 5 + 1 = 11

4th Iteration:

S -> C -> B -> D -> F -> G = g(n) + h(n) = 1 + 3 + 2 + 5 + 1 + 0 = 11 (goal state reached)

From all the above calculations, we can see that S -> C -> F -> G gives the minimum cost path to reach goal state. Hence A\* Algorithm chooses that path.

**Q.4 (b) Explain a partial order planner with an example.**

Partial-Order Planner(POP) is a regression planner; it uses problem decomposition; it searches plan space rather than state space; it build partially-ordered plans; and it operates by the principle of least-commitment.

A plan in POP (whether it be a finished one or an unfinished one) comprises:

* A set of plan steps. Each of these is a STRIPS operator, but with the variables instantiated.
* A set of ordering constraints: Si -< Sj means step Si must occur sometime before Sj (not necessarily immediately before).
* A set of causal links: Sic→SjSic→Sj means step Si achieves precondition c of step Sj .

So, it comprises actions (steps) with constraints (for ordering and causality) on them.

The algorithm needs to start off with an initial plan. This is an unfinished plan, which we will refine until we reach a solution plan. The initial plan comprises two dummy steps, called Start and Finish.

Start is a step with no preconditions, only effects: the effects are the initial state of the world. Finish is a step with no effects, only preconditions: the preconditions are the goal.

Take example of wearing shoe to understand partial ordered planning.

A partial order planning combines two action sequences

* First branch covers left-sock and left-shoe.
* In this case to wear a left shoe, wearing left sock is the precondition, similarly.
* Second branch covers right-sock and right-shoe.
* Here, wearing a right sock is the precondition for wearing the right shoe.

Once these actions are taken we achieve our goal and reach the finish state.