1a.

Sate configuration: [{selected coins}, {remaining coins in descending order}, remaining value, h(n), g(n), f(n)]; sum of selected coins <= C

Transformation operator: select a coin. Each coin selection incurs a cost 1

Heuristic function: $h(n) =$ total max no of coins with added value (when considered in the descending order of their denominations) less than or equal the remaining value

Initial state: [{}, {8,7,6,4,2,1}, 13, 1, 0, 1]

Goal State: remaining value = 0

1b.

OPEN = [{}, {8,7,6,4,2,1}, 13, 1, 0, 1]

CLOSED = []

OPEN = **[{8}, {7,6,4,2,1}, 5, 0, 1, 1]**, [{7}, {8,6,4,2,1}, 6, 0, 1, 1], [{6}, {8,7,4,2,1}, 7, 0,1,1], [{4}, {8,7,6,2,1}, 9, 1, 1, 2], [{2}, {8,7,6,4,1}, 11, 1, 1, 2], [{1}, {8,7,6,4,2}, 12, 1, 1, 2]

CLOSED = [{}, {8,7,6,4,2,1}, 13, 1, 0, 1]

OPEN = **[{7}, {8,6,4,2,1}, 6, 0, 1, 1]**, [{6}, {8,7,4,2,1}, 7, 0,1,1], [{4}, {8,7,6,2,1}, 9, 1, 1, 2], [{2}, {8,7,6,4,1}, 11, 1, 1, 2], [{1}, {8,7,6,4,2}, 12, 1, 1, 2], [{8,4}, {7,6,2,1}, 1, 0, 2, 2], [{8,2}, {7,6,4,1}, 3, 0, 2, 2], [{8,1}, {7,6,4,2}, 4, 0, 2,2]

CLOSED = [{}, {8,7,6,4,2,1}, 13, 1, 0, 1], [{8}, {7,6,4,2,1}, 5, 0, 1, 1]

OPEN = **[{6}, {8,7,4,2,1}, 7, 0,1,1]**, [{4}, {8,7,6,2,1}, 9, 1, 1, 2], [{2}, {8,7,6,4,1}, 11, 1, 1, 2], [{1}, {8,7,6,4,2}, 12, 1, 1, 2], [{8,4}, {7,6,2,1}, 1, 0, 2, 2], [{8,2}, {7,6,4,1}, 3, 0, 2, 2], [{8,1}, {7,6,4,2}, 4, 0, 2,3], [{7,6}, {8,4,2,1}, 0, 0, 2, 2], [{7,4}, {8,6,2,1}, 2, 0, 2, 2], [{7,2}, {8,6,4,1}, 4, 0, 2, 2], [{7,1}, {8,6,4,2}, 5, 0, 2,2]

CLOSED = [{}, {8,7,6,4,2,1}, 13, 1, 0, 1], [{8}, {7,6,4,2,1}, 5, 0, 1, 1], [{7}, {8,6,4,2,1}, 6, 0, 1, 1]

Continue till the Goal is in the selected to be put in the CLOSED list.

Another state-space formulation:
State configuration remains the same
Operator: Either select the highest feasible denomination of the remaining coins (cost 1) or not (cost 0)
Heuristics: As formulation 1
Initial state: [{}, {8,7,6,4,2,1}, 13, 1, 0, 1]
Goal state: remaining C = 0

OPEN = **[{}, {8,7,6,4,2,1}, 13, 1, 0, 1]**
CLOSED = []

OPEN = **[{8}, {7,6,4,2,1}, 5, 0, 1, 1]**, [{}, {7,6,4,2,1}, 13, 2, 0, 2]
CLOSED = [{}, {8,7,6,4,2,1}, 13, 1, 0, 1]

OPEN = [{}, {7,6,4,2,1}, 13, 2, 0, 2], **[{8,4}, {2,1}, 1, 0, 2, 2]**
CLOSED = [{}, {8,7,6,4,2,1}, 13, 1, 0, 1], [{8}, {7,6,4,2,1}, 5, 0, 1, 1]

OPEN = [{8,4,1}, {}, 0, 0, 3, 3], [{8,4}, {}, 1, 0, 2, 2], **[{}, {7,6,4,2,1}, 13, 2, 0, 2]**
CLOSED = [{}, {8,7,6,4,2,1}, 13, 1, 0, 1], [{8}, {7,6,4,2,1}, 5, 0, 1, 1], [{8,4}, {2,1}, 1, 0, 2, 2]

OPEN = [{8,4,1}, {}, 0, 0, 3, 3], [{8,4}, {}, 1, 0, 2, 2], **[{7}, {6,4,2,1}, 6, 1, 1, 2]**,[{}, {6,4,2,1}, 13, 4,0,4]
CLOSED = [{}, {8,7,6,4,2,1}, 13, 1, 0, 1], [{8}, {7,6,4,2,1}, 5, 0, 1, 1], [{8,4}, {2,1}, 1, 0, 2, 2], [{}, {7,6,4,2,1}, 13, 2, 0, 2]

OPEN = [{8,4,1}, {}, 0, 0, 3, 3], [{8,4}, {}, 1, 0, 2, 2], [{}, {6,4,2,1}, 13, 4,0,4], **[{7,6}, {4,2,1}, 0, 0, 2, 2]**, [{7}, {4,2,1}, 6, 2, 1, 3]
CLOSED = [{}, {8,7,6,4,2,1}, 13, 1, 0, 1], [{8}, {7,6,4,2,1}, 5, 0, 1, 1], [{8,4}, {2,1}, 1, 0, 2, 2], [{}, {7,6,4,2,1}, 13, 2, 0, 2], [{7}, {6,4,2,1}, 6, 1, 1, 2]

Goal Node **[{7,6}, {4,2,1}, 0, 0, 2, 2]** is in the CLOSED list in the next step.

The state space in this formulation is expected to be smaller than the earlier one.

Other heuristics formulations:

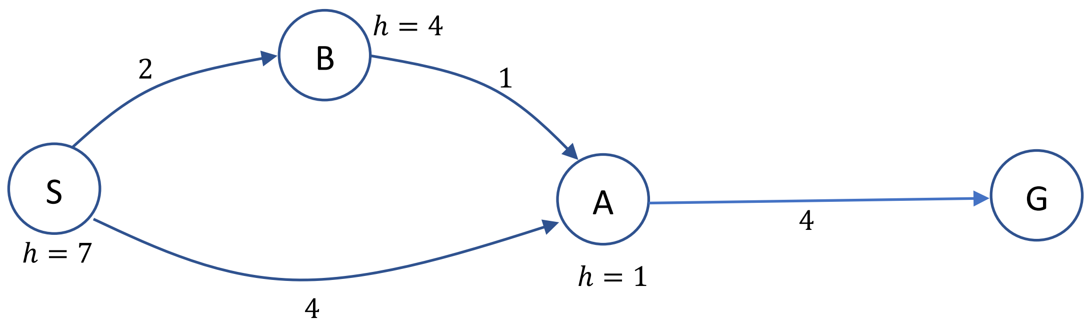
$h(n) = \frac{C - \sum selected\ coin\ values\ at\ state\ n}{\sum Remaining\ coin\ values}$ [a grossly underestimated heuristic func. It may lead to large no of node expansions and inefficient search]

$h(n) = \frac{C - \sum selected\ coin\ values\ at\ state\ n}{No\ of\ remaining\ coins}$ [may lead to inadmissible heuristic function. It may not guarantee optimal solution]

2a.

We are to perform A* on a state space graph. That means, for the guarantee of optimality, the heuristic function should be admissible (underestimated at each node) and consistent $c(n_i, n_j) \geq n_i - n_j, n_j$ being the child of n_i.

If we can find at least one case, where the heuristic value at each node is underestimated and optimal solution could not be found we can prove the statement to be true.



OPEN = [<S,7>]

CLOSED = []

OPEN = [<A,5>, <B,6>]

CLOSED = [<S,7>]

OPEN = [<G,8>, <B,6>]

CLOSED = [<A,5>,<S,7>]

Now we should expand node B having child node A. As we have closed node A by earlier path, we will not revisit it.

OPEN = [<G,8>]

CLOSED = [<B,6><A,5>,<S,7>]

OPEN = []

CLOSED = [<G,8><B,6><A,5>,<S,7>]

The solution cost from S →A→ G is 8 which is not optimal.

Thus, the optimal cost solution may sometimes NOT be found when the heuristic estimate at every node are underestimated is a TRUE statement.

One may observe here that $h(S) - h(A) = 6 - 1 = 5 < c(S, A) = 4$. Thus, the heuristic function is not consistent in this case leading to sub-optimal solution

Let us try to make the heuristic value at node A consistent by setting it to $h(A) = 3$. Now, A* will follow the steps below:

OPEN = [<S,7>]

CLOSED = []

OPEN = [<A,7>, <B,6>]

CLOSED = [<S,7>]

Node A, the child of B will be considered with cost (g=3, h=3, f=6). As A is in the OPEN list, the f value will be updated to 6 as it is lower than 7. The parent of node A will be changed to node B.

OPEN = [<A,6>]

CLOSED = [<B,6>,<S,7>]

OPEN = [<G,7>]

CLOSED = [<A,6><B,6>,<S,7>]

OPEN = []

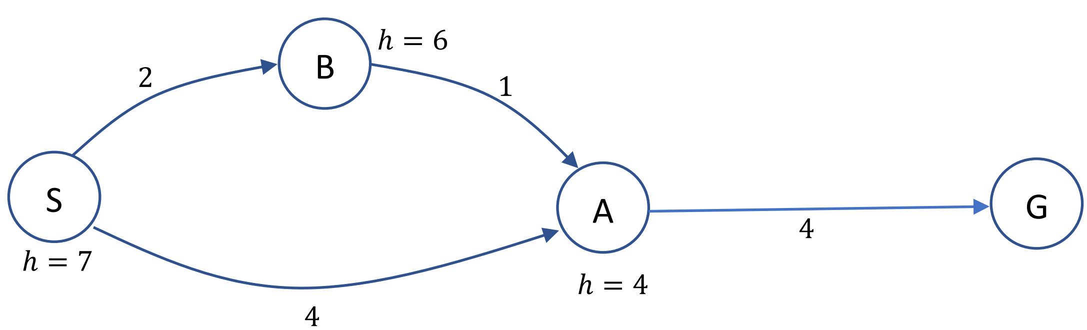
CLOSED = [<G,7>,<A,6><B,6>,<S,7>]

Now, we observe that A* terminates with an optimal solution S→B→A→G when the heuristic function is underestimated and consistent.

2b.

Only one non-goal node is overestimated ➔ Optimal cost solution guaranteed NOT to be found

Let us consider a case where one non-goal node is overestimated. If we can show that we may find the optimal solution, then the statement may be contradicted.



Here, heuristic value at node B is overestimated.

OPEN = [<S,7>], CLOSED = []

OPEN = [<A,8>,<B,8>], CLOSED = [<S,7>]

OPEN = [<A,8/7>], CLOSED = [<B,8>] → A is in open and has a lower f value through another path

OPEN = [<G,7>], CLOSED = [<B,8>, <A,7>]

OPEN = [], CLOSED = [<B,8>, <A,7>, <G,7>]

So, node B will be selected for expansion leading to the solution S→B→A→G, with optimal cost.

Optimal cost solution is guaranteed NOT to be found when only one non-goal node is overestimated is a FALSE statement.

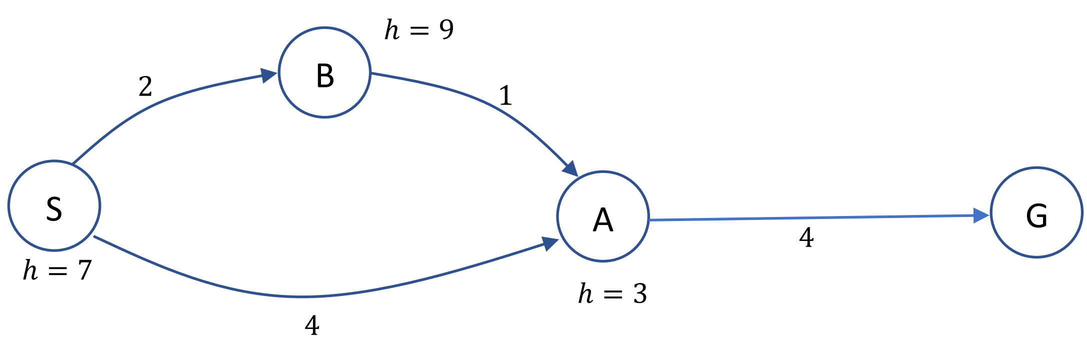
2c

Only one non-goal node is overestimated ➔ Optimal cost solution NOT guaranteed to be found

We need to show that when only a single non-goal node is overestimated, sometimes optimal solution can be found; sometimes an optimal solution cannot be found.

Optimal solution can be found has been illustrated with example 2b

Showing other case:



Here, heuristic value at node B is overestimated.

OPEN = [<S,7>], CLOSED = []

OPEN = [<A,7>, <B, 11>], CLOSED = [<S,7>]

OPEN = [<B,11>, <G,8>], CLOSED = [<A,7>, <S,7>]

OPEN = [<B,11>], CLOSED = [<G,8>,<A,7>,<S,7>]

So, node A will be selected for expansion leading to the solution S→A→G, with non-optimal cost.

Optimal cost solution is NOT guaranteed to be found when only one non-goal node is overestimated is a TRUE statement.