**UIT2511---Software Development Project – II**

**Exercise — 03**

**Name:** B.Vasundhara

**Roll No:** 3122 21 5002 119

1. **Formulate the above version of the block’s world as a search problem, by precisely defining the state space, the initial state, the goal test, the set of possible actions and the path cost.**

List of predicates:

* ON(A,B) : Block A is on B
* ONTABLE(A) : A is on table
* CLEAR(A) : Nothing is on top of A
* HOLDING(A) : Arm is holding A.
* ARMEMPTY : Arm is holding nothing

The Robot Arm can perform 4 operations:

* STACK(X,Y) : Stacking Block X on Block Y
* UNSTACK(X,Y) : Picking up Block X which is on top of Block Y
* PICKUP(X) : Picking up Block X which is on top of the table
* PUTDOWN(X) : Put Block X on the table

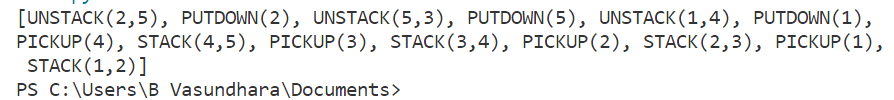
A white background with black text

Description automatically generated

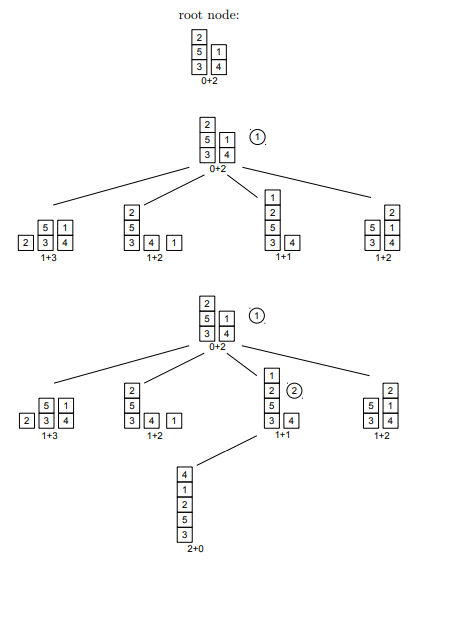
A close-up of a number

Description automatically generated

To achieve the goal state:



1. **Assume that the initial configuration is the one shown on the left in the figure above, and consider a heuristic function defined as the number of blocks that are not in the highest pile (or in one of the highest piles). Under this setting, show how the A\* search strategy expands the first four nodes of the search tree, avoiding repeated states. As in the previous exercise, when drawing the search tree, you should clearly indicate: the order of expansion of each node; the action corresponding to each edge of the tree; the state, the path cost and the value of the heuristic of each node.**



**Search Tree:**

1. **Node 1 (Root Node)**
   * State: Same as the initial configuration.
   * Path Cost: 0 (Starting node).
   * Heuristic Value: 2 (Blocks '1' and '2' are not in the highest pile).
   * Expansion Order: 1
   * Actions:
     + No action (Initial state).
2. **Node 2 (Expand Node 1)**
   * State: Block '1' is stacked on top of Block '4', other blocks remain the same.
   * Path Cost: 1 (One action to stack '1' on '4').
   * Heuristic Value: 1 (Block '2' is not in the highest pile).
   * Expansion Order: 2
   * Actions:
     + STACK('1', '4')
3. **Node 3 (Expand Node 1)**
   * State: Block '2' is stacked on top of Block '5', other blocks remain the same.
   * Path Cost: 1 (One action to stack '2' on '5').
   * Heuristic Value: 1 (Block '1' is not in the highest pile).
   * Expansion Order: 3
   * Actions:
     + STACK('2', '5')
4. **Node 4 (Expand Node 2)**
   * State: Block '1' is on top of Block '4', other blocks remain the same.
   * Path Cost: 2 (One action to unstack '1' from '4' and one action to stack it on '5').
   * Heuristic Value: 1 (Block '2' is not in the highest pile).
   * Expansion Order: 4
   * Actions:
     + UNSTACK('1', '4')
     + STACK('1', '5')
5. **Define another possible admissible heuristic and prove its admissibility.**

The proposed heuristic for the Block's World problem is defined as follows:

Let 'n' be the number of blocks on top of which there is a block different from the one in the goal state. The heuristic is calculated as ⌊n/2⌋.

For example, if the goal state is {(1, 2, 3, 4, 5)}, and the current state is {(2, 5, 3), (1, 4)}, you can calculate the heuristic as follows:

- Block 1 has no block on top of it, as in the goal state, so it counts as 0.

- Block 2 has a different block (5) on top of it, contrary to the goal state, so it counts as 1.

- Block 5 has a different block (2) on top of it, contrary to the goal state, so it counts as 1.

- Block 3 has a different block (1) on top of it, contrary to the goal state, so it counts as 1.

- Block 4 has no block on top of it, as in the goal state, so it counts as 0.

Now, you calculate ⌊n/2⌋, where n is the total count of blocks that have a different block on top of them:

⌊4/2⌋ = 2

This means that at least two blocks must be moved to reach the goal state from the given state.

The heuristic is admissible because, by moving one block from any state 's', in the resulting state 's'', at most two other blocks will have a different block on top of them compared to 's'. Therefore, at most two more blocks than in 's' can have the correct block on top of them, as in the goal state. This heuristic provides a lower bound on the number of moves required to reach the goal state, making it a valid admissible heuristic.