**Question 1**

|  |  |  |
| --- | --- | --- |
| Search Strategy | Advantage | Disadvantage |
| Depth-first Search | Requires relatively little memory space | Will go into an infinite loop if the search space is cyclical |
| Depth-first search with iterative deepening | Requires relatively little memory space | Needs to recompute many paths when depth limit is increased |
| Breadth-first search |  | Stores many possible paths and can thus require a lot of space |
| A\* | Will find a reasonable solution but not necessarily the best solution. | Has to keep a number of candidate nodes in memory and also keep their heuristic function in memory |

**Question 2**

**Question 3.1**

S=[clear(a).clear(b), clear(c),clear(d),on(a,1),on(b,2),on(c,3),on(d,4)]

Our Goal: on(d,1)

On(d,1) does not hold so we propose move(d,From,1)

Preconditions:[clear(d),clear(1),on(d,From)]

clear(d) holds

clear(1) does not hold

new goal clear(1)

Propose action: move(Block,1,To)

Preconditions:[clear(Block),clear(To),on(Block,1)]

clear(Block) holds for Block = a

clear(To) holds for To = b

on(Block,1) holds for Block = a

Action move(a,1,b)

After deletions S=[clear(a), clear(c),clear(d),on(b,2),on(c,3),on(d,4)]

After additions S=[clear(a), clear(c),clear(d), clear(1), on(b,2),on(c,3),on(d,4),on(a,b)]

Back to goal on(d,1)

Proposed move(d,From,1)

Preconditions: [clear(d),clear(1),on(d,From)]

clear(d) holds

clear(a) holds

on(d,From) holds for From = 4

Action move(d,4,1)

After deletions S=[clear(a), clear(c),clear(d), on(b,2),on(c,3),on(a,b)]

After additions S=[clear(a), clear(c),clear(d),clear(4), on(b,2),on(c,3),on(d,1),on(a,b)]

Plan:[move(a,1,b) move(d,4,1) ]

**Question 3.2**

S=[clear(a).clear(b), clear(c),clear(d),on(a,1),on(b,2),on(c,3),on(d,4)]

Our Goal: on(d,1)

On(d,1) does not hold so we propose move(d,From,1)

Preconditions:[clear(d),clear(1),on(d,From)]

clear(d) holds

clear(1) does not hold

new goal clear(1)

Propose action: move(Block,1,To)

Preconditions:[clear(Block),clear(To),on(Block,1)]

clear(Block) holds for Block = a

clear(To) holds for To = b

on(Block,1) holds for Block = a

Action move(a,1,b)

After deletions S=[clear(a), clear(c),clear(d),on(b,2),on(c,3),on(d,4)]

After additions S=[clear(a), clear(c),clear(d), clear(1), on(b,2),on(c,3),on(d,4),on(a,b)]

Protected goal list: [clear(1)]

Back to goal on(d,1)

Proposed move(d,From,1)

Preconditions: [clear(d),clear(1),on(d,From)]

clear(d) holds

clear(a) holds

on(d,From) holds for From = 4

Action move(d,4,1)

After deletions S=[clear(a), clear(c),clear(d), on(b,2),on(c,3),on(a,b)]

After additions S=[clear(a), clear(c),clear(d),clear(4), on(b,2),on(c,3),on(d,1),on(a,b)]

Plan:[move(a,1,b) move(d,4,1) ]

**Question 4**