3a

Output:

Enter graph: {'A': {'B': 1, 'C': 4}, 'B': {'A': 1, 'C': 2, 'D': 5}, 'C': {'A': 4, 'B': 2, 'D': 1}, 'D': {'B': 5, 'C': 1}}

Enter start node: A

Enter goal node: D

F: ['A']

E: set()

Node: A Parent: None Cost: 0

F: ['B', 'C']

E: {'A'}

Node: B Parent: A Cost: 1

F: ['C', 'D', 'B']

E: {'A', 'B'}

Node: C Parent: B Cost: 3

F: ['D', 'C', 'B']

E: {'A', 'B', 'C'}

Node: D Parent: C Cost: 4

Goal node found

3b

Output:

Enter the graph: {'A': ['B', 'C'], 'B': ['D', 'E'], 'C': ['F'], 'D': [], 'E': ['F'], 'F': []}

Enter Start node: A

Enter goal node: F

Enter max Depth: 3

f: ['A']

e: []

Node: A Parent: None Depth: 0

Goal node not found within depth limit

f: ['A']

e: ['A']

Node: A Parent: None Depth: 0

Node: B Parent: A Depth: 1

Node: C Parent: A Depth: 1

Goal node not found within depth limit

f: ['A']

e: ['A', 'B', 'C']

Node: A Parent: None Depth: 0

Node: B Parent: A Depth: 1

Node: D Parent: B Depth: 2

Node: E Parent: B Depth: 2

Node: C Parent: A Depth: 1

Node: F Parent: C Depth: 2

Goal node found

3.c

Output:

Enter the graph: {'A': ['B', 'C'], 'B': ['D', 'E'], 'C': ['F'], 'D': [], 'E': ['F'], 'F': []}

Enter Start node: A

Enter goal node: F

F\_Front [['A', None]]

E\_Front []

F\_Back [['F', None]]

E\_Back []

F\_Front [['B', 'A'], ['C', 'A']]

E\_Front [['A']]

F\_Back [['F', None]]

E\_Back [['F', None]]

F\_Front [['C', 'A'], ['B', 'A']]

E\_Front [['A']]

F\_Front [['B', 'A'], ['C', 'A']]

E\_Front [['A']]

Goal node found

The Path from A to F is: [['A', None], ['B', 'A'], ['D', 'B']]

The search intersected at: D

4.a

Output:

Enter Big jug capacity: 4

Enter Small jug capacity: 3

Enter t jug capacity: 2

f: [(0, 0)]

e: []

f: [(4, 0)]

e: [(0, 0)]

f: [(1, 3)]

e: [(0, 0), (4, 0)]

f: [(1, 0)]

e: [(0, 0), (4, 0), (1, 3)]

f: [(4, 0)]

e: [(0, 0), (4, 0), (1, 3), (1, 0)]

f: [(2, 3)]

e: [(0, 0), (4, 0), (1, 3), (1, 0), (4, 0)]

t achieved: [(0, 0), (4, 0), (1, 3), (1, 0), (4, 0), (2, 3)]

4.b

Output:

Enter Big jug capacity: 4

Enter Small jug capacity: 3

Enter t jug capacity: 2

f: [(0, 0)]

e: []

f: [(4, 0)]

e: [(0, 0)]

f: [(1, 3)]

e: [(0, 0), (4, 0)]

f: [(1, 0)]

e: [(0, 0), (4, 0), (1, 3)]

f: [(4, 0)]

e: [(0, 0), (4, 0), (1, 3), (1, 0)]

f: [(2, 3)]

e: [(0, 0), (4, 0), (1, 3), (1, 0), (4, 0)]

t achieved: [(0, 0), (4, 0), (1, 3), (1, 0), (4, 0), (2, 3)]

5.a

Output:

f: [(3, 3, 0, 0, 0)]

e: []

f: [(2, 3, 1, 0, 1)]

e: [(3, 3, 0, 0, 0)]

f: [(2, 2, 1, 1, 1)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1)]

f: [(2, 2, 2, 0, 0)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1)]

f: [(1, 2, 2, 1, 0)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0)]

f: [(1, 1, 2, 2, 0)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0)]

f: [(1, 1, 3, 0, 1)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0)]

f: [(0, 1, 3, 1, 1)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0), (1, 1, 3, 0, 1)]

f: [(0, 0, 3, 2, 1)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0), (1, 1, 3, 0, 1), (0, 1, 3, 1, 1)]

f: [(0, 0, 3, 3, 0)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0), (1, 1, 3, 0, 1), (0, 1, 3, 1, 1), (0, 0, 3, 2, 1)]

Crossed Successfully...

Path: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0), (1, 1, 3, 0, 1), (0, 1, 3, 1, 1), (0, 0, 3, 3, 0)]

5.b

Outut:

f: [(3, 3, 0, 0, 0)]

e: []

f: [(2, 3, 1, 0, 1)]

e: [(3, 3, 0, 0, 0)]

f: [(2, 2, 1, 1, 1)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1)]

f: [(2, 2, 2, 0, 0)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1)]

f: [(1, 2, 2, 1, 0)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0)]

f: [(1, 1, 2, 2, 0)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0)]

f: [(1, 1, 3, 0, 1)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0)]

f: [(0, 1, 3, 1, 1)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0), (1, 1, 3, 0, 1)]

f: [(0, 0, 3, 2, 1)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0), (1, 1, 3, 0, 1), (0, 1, 3, 1, 1)]

f: [(0, 0, 3, 3, 0)]

e: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0), (1, 1, 3, 0, 1), (0, 1, 3, 1, 1), (0, 0, 3, 2, 1)]

Crossed Successfully...

Path: [(3, 3, 0, 0, 0), (2, 3, 1, 0, 1), (2, 2, 1, 1, 1), (2, 2, 2, 0, 0), (1, 2, 2, 1, 0), (1, 1, 2, 2, 0), (1, 1, 3, 0, 1), (0, 1, 3, 1, 1), (0, 0, 3, 3, 0)]