Question 1

Implementations is submitted as search.java

Question 2

(note: DepthFirstTreeSearch GreedyBestFirstTreeSearch GreedyBestGraphSearch are not completed, luckily I got the solution in some trying with more cost.)

Wolf Goat Cabbage Problem:

TreeSearch——————————
BreadthFirstTreeSearch: (cost=7.0, expansions=202)
UniformCostTreeSearch: (cost=7.0, expansions=167)
DepthFirstTreeSearch: (cost=241.0, expansions=241)
GreedyBestFirstTreeSearch: (cost=129.0, expansions=258)
AstarTreeSearch: (cost=7.0, expansions=246)

GraphSearch—————————-
BreadthFirstGraphSearch: (cost=7.0, expansions=14)
UniformCostGraphSearch: (cost=7.0, expansions=14)
DepthFirstGraphSearch: (cost=7.0, expansions=8)
GreedyBestGraphSearch: (cost=7.0, expansions=12)
AstarGraphSearch: (cost=7.0, expansions=14)

IterativeDeepening—————————-
IterativeDeepeningTreeSearch: (cost=7.0, expansions=72)
IterativeDeepeningGraphSearch: (cost=7.0, expansions=10)

Using heuristic function that h(s) = the total number of wolf, goat and cabbage that are not in the desination, which is a admissible heuristic function.
AstarTreeSearch: (cost=7.0, expansions=121)
AstarGraphSearch: (cost=7.0, expansions=14)

Question 3

Missionaries and cannibals problem:

TreeSearch——————————

BreadthFirstTreeSearch: (cost=9.0, expansions=907)
UniformCostTreeSearch: (cost=9.0, expansions=667)
DepthFirstTreeSearch: (cost=33.0, expansions=33)
GreedyBestFirstTreeSearch: (cost=103.0, expansions=206)
AstarTreeSearch: (cost=9.0, expansions=958)

GraphSearch———————-
BreadthFirstGraphSearch: (cost=9.0, expansions=13)
UniformCostGraphSearch: (cost=9.0, expansions=13)
DepthFirstGraphSearch: (cost=9.0, expansions=12)
GreedyBestGraphSearch: (cost=9.0, expansions=13)
AstarGraphSearch: (cost=9.0, expansions=13)

IterativeDeepening———————-
IterativeDeepeningTreeSearch: (cost=9.0, expansions=36)
IterativeDeepeningGraphSearch: (cost=9.0, expansions=10)

Using heuristic function that h(s) = the total number of missionaries and cannibals that are not on the destination bank, which is a admissible heuristic function.
AstarTreeSearch: (cost=9.0, expansions=95)
AstarGraphSearch: (cost=9.0, expansions=12)

Question 4

Water Jugs Problem:
TreeSearch————————
BreadthFirstTreeSearch: (cost=3.0, expansions=101)
UniformCostTreeSearch: (cost=3.0, expansions=86)
DepthFirstTreeSearch: (cost=34.0, expansions=34)
GreedyBestFirstTreeSearch: (cost=72.0, expansions=144)
AstarTreeSearch: (cost=3.0, expansions=48)

GraphSearch———————-
BreadthFirstGraphSearch: (cost=3.0, expansions=28)
UniformCostGraphSearch: (cost=3.0, expansions=16)
DepthFirstGraphSearch: (cost=23.0, expansions=23)
GreedyBestGraphSearch: (cost=12.0, expansions=19)

AstarGraphSearch: (cost=3.0, expansions=13)

IterativeDeepening—————————-
IterativeDeepeningTreeSearch: (cost=3.0, expansions=5)
IterativeDeepeningGraphSearch: (cost=3.0, expansions=9)

Question 5

Pancake Sorting Problem
TreeSearch——————————
BreadthFirstTreeSearch: (cost=5.0, expansions=6455)
UniformCostTreeSearch: (cost=5.0, expansions=4996)
DepthFirstTreeSearch: (cost=3608.0, expansions=3608)
GreedyBestFirstTreeSearch: (cost=637.0, expansions=1273)
AstarTreeSearch: (cost=5.0, expansions=7382)

GraphSearch—————————-
BreadthFirstGraphSearch: (cost=5.0, expansions=444)
UniformCostGraphSearch: (cost=5.0, expansions=460)
DepthFirstGraphSearch: (cost=572.0, expansions=614)
GreedyBestGraphSearch: (cost=250.0, expansions=389)
AstarGraphSearch: (cost=5.0, expansions=535)

IterativeDeepening—————————-
IterativeDeepeningTreeSearch: (cost=5.0, expansions=1267)
IterativeDeepeningGraphSearch: (cost=5.0, expansions=166)

Using the heuristic function to count the pancake not in position:
AstarTreeSearch: (cost=5.0, expansions=48)
AstarGraphSearch: (cost=5.0, expansions=19)