Part 1

Question1 Write the description for the initial and goal states using PDDL.

Initial states:
At(Monkey,A)∧At(Box,B)∧At(Bananas,C)∧
Height(Monkey,Low)∧Height(Box,Low)∧Height(Banana,Height)

Goal states:
Have(Monkey,Banana)

Question2
Write all the action PDDL schemas (there are 6 actions in total). Each action should include a name, variables, precondition and effect. The precondition of each action should be set in a way that self-connection is avoided, i.e. no action connects any state to itself.

Action(GO(x, y),
Precond: At(Monkey, x) ∧ ¬ Height(Monkey,High),
Effect: At(Monkey, y) ∧ ¬ At(Monkey, x))

Action(Push(box, x, y),
Precond: At(Monkey, x) ∧ At(box, x),
Effect: At(Monkey, y) ∧ At(box, y) ∧ ¬ At(Monkey, x) ∧ ¬ At(box, y))

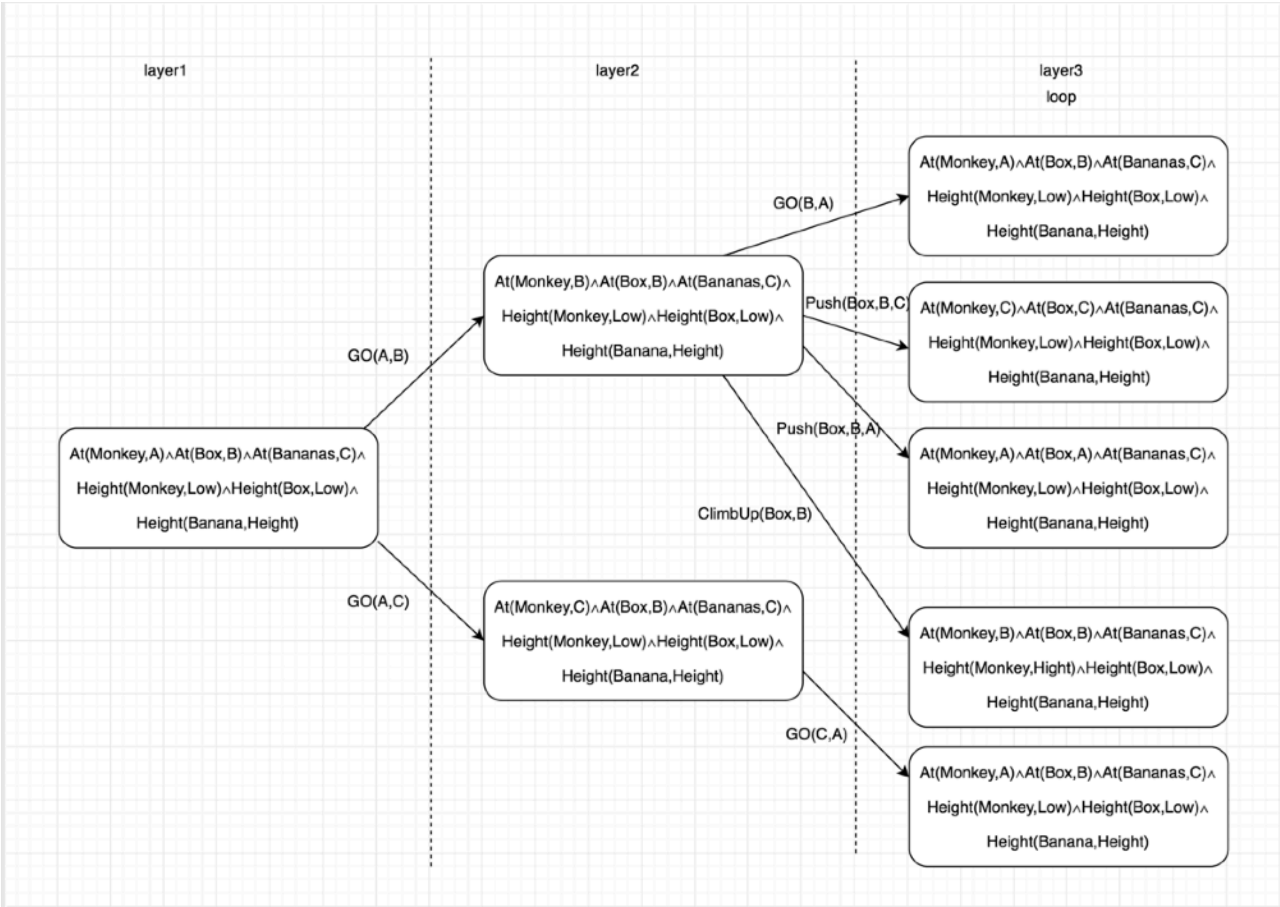
Action(ClimbUp(box,x),
Precond: At(Monkey, x) ∧ At(box, x),
Effect: Height(Monkey, Hight) ∧ ¬ Height(Monkey, Low))

Action(ClimbDown(box, x, y),
Precond: At(Monkey, x) ∧ At(box, x) ∧ Height(Monkey, High)
Effect: Height(Monkey, Low) ∧ ¬ Height(Monkey, High))

Action(Grasp(monkey,x,h),
Precond: At(Monkey, x) ∧ At(box, x) ∧ Height(Monkey, h) ∧ Height(Banana, h)
Effect: Have(Monkey,Banana))

Action(UnGrasp(banana),
Precond: Have(Monkey,banana)
Effect: ¬ Have(Monkey,banana)

Question3
A plan to achieve the goal state from the initial state can be found by using forward state-space search. Based on the PDDL in Questions 1 and 2, draw the first three layers of the corresponding state-space search graph to demonstrate the search process.



4 Write a plan to achieve the goal state from the initial state. The plan needs to be formatted as follows:

Initial state:
At(Monkey,A)∧At(Box,B)∧At(Bananas,C)∧
Height(Monkey,Low)∧Height(Box,Low)∧Height(Banana,Height)

Action(GO(A, B))

State:
At(Monkey,B)∧At(Box,B)∧At(Bananas,C)∧
Height(Monkey,Low)∧Height(Box,Low)∧Height(Banana,Height)

Action(Push(box, B, C))

State:
At(Monkey,C)∧At(Box,C)∧At(Bananas,C)∧
Height(Monkey,Low)∧Height(Box,Low)∧Height(Banana,Height)

Action(ClimbUp(box,x))

State:
At(Monkey,C)∧At(Box,C)∧At(Bananas,C)∧
Height(Monkey,Height)∧Height(Box,Low)∧Height(Banana,Height)

Action(Grasp(Monkey,C,Height),

Gold State:
At(Monkey,C)∧At(Box,C)∧At(Bananas,C)∧
Height(Monkey,Height)∧Height(Box,Low)∧Height(Banana,Height)∧Have(Banana)

Part 2

Question1

Given a schedule whose action sequence is as follows: P rocess(O_{11}, M_1, t_1) → P rocess(O_{21}, M_2, t_2) → Process(O_{31},M_1,t_3) → Process(O_{12},M_2,t_4) → Process(O_{22},M_1,t_5) → Process(O_{32},M_2,t_6). Since the se- quence is sorted in the increasing order of starting time, we know that $t_1 \le t_2 \le t_3 \le t_4 \le t_5 \le t_6$. Calculate the earliest starting time (t_1 to t_6) of each action. You can draw gantt chart to help you think. Hint: the earliest starting time of an action is the later time between the earliest ready time of the operation and the earliest idle time of the machine.

t1=0, t2= 10, t3=50, t4=50, t5=90, t6=90

Question2

For the solution given in Question 1, find the completion time of each job, which is the finishing time of its last operation. Then, calculate the makespan of the solution, which is defined as the maximum completion time of all the jobs.

Finishing time of J1 = t4 + Proc(O12) = 75
Finishing time of J2 = t5 + Proc(O22) = 125
Finishing time of J3 = t6 + Proc(O32) = 110
makespan of the solution which is the finishing time of its last operation: 125

Question3

Write the state from step 1 to step 3, and the final solution when applying the Shortest Processing time (SPT) dispatching rule to the problem. At each step, the representation of a state is composed of (1) a partial solution, (2) the earliest idle time of each machine and (3) the earliest ready time of each unprocessed operation. The initial state (step 0) is given below for your reference.

Step1:
Partial solution: Process(0_{11},M1,0)
earliestIdleTime(M_1) =50, earliestIdleTime(M_2) =0
earliestReadyTime(0_{12})= 50
earliestReadyTime(0_{21})= 10 earliestReadyTime(0_{22})= ∞
earliestReadyTime(0_{31})= 20 earliestReadyTime(0_{32})= ∞

Step2:
Partial solution: Process(0_{11},M1,0)->Process(0_{21},M2,10)
earliestIdleTime(M_1) =50, earliestIdleTime(M_2) =40
earliestReadyTime(0_{12})= 50
earliestReadyTime(0_{22})= 40
earliestReadyTime(0_{31})= 20 earliestReadyTime(0_{32})= ∞

Step3:
Partial solution:Process(0_{11},M1,0)->Process(0_{21},M2,10)->Process(0_{12},M2,50)
earliestIdleTime(M_1) =50, earliestIdleTime(M_2) =75
earliestReadyTime(0_{22})= 40
earliestReadyTime(0_{31})= 20 earliestReadyTime(0_{32})= ∞

Final solution:
Process(0_{11},M1,0)->Process(0_{21},M2,10)->Process(0_{12},M2,50)>Process(0_{22},M1,50) ->Process(0_{31},M1,85) ->Process(0_{32},M2,125)

Question4
For the solution obtained by the SPT rule, calculate the completion time of each job and the makespan. Compare the makespan between this solution with that obtained in Question 1 to find out which solution is better in makespan. Note: the solution in Question 1 is obtained by the First-Come-First-Serve (FCFS) rule.

Use SPT rule:
Finish time of job1:
starting time of O_{12}+ProcTime(O_{12}) =
50+25=75

Finish time of job2:
starting time of O_{22}+ProcTime(O_{22}) =
50+35=85

Finish time of job3:
starting time of O_{32}+ProcTime(O_{32}) =
125+20=145

The makespan of the solution is the time of completion time of latest job is 145

Use FCFS rule:
completion time of job1:
starting time of O_{12}+Proc(O12) =
50+25=75

completion time of job2:
starting time of O_{22}+Proc(O22) =
90+35=125

completion time of job3:
t6+Proc(O32) =
90+20=110

The makespan of the solution is the time of completion time of latest job is 125
compare both solutions above, FCFS rule has better performance.

Question 5
The two compared solutions are obtained by the SPT and FCFS rules, respectively. If one solution is better then the other, does it mean that the rule that generates the better solution is better than the other rule? Why?

No, we can not conclude that the rule that generates the better solution is better than the other rule. Because during the calculation process, Use SPT rule to calculate the completion time of job 2 is have better performance. I think they can not compare depends on the solution, but they can compare depends on the job.(like SPT rule is doing better then FCFS rules in job 2).

Part3
Question1
Find a solution by the nearest neighbour heuristic. Write the solution as a set of node sequences starting and ending at the depot node (node 1). It should look like R_1 = (1,...,1), R_2 = (1,...,1), ...

R1(1,2,3,5,1)
R2 (1,6,8,4,1)
R3 (1,7,9,10,1)

Question2.
Calculate the total length (Euclidean distance) for the obtained solution.

len(R1)
= len(1,2)+ len(2,3)+ len(3,5)+ len(5,1)
= 1+1+1+2.24=5.24

len(R2)
= len(1,6)+ len(6,8)+ len(8,4)+ len(4,1)
=1.41+1.41+1.41+3.16
=7.39

len(R3)
= len(1,7)+ len(7,9)+ len(9,10)+ len(10,1)
=2.23+3.16+2+5.39
=12.78

total length
= len(R1) + len(R2) + len(R3)
= 5.24+7.39+12.78
=25.41