COMP 307 Assignment 4.

Part 1.

(a) Q_1:

Initial state:

At(Monkey, A) ∧ At(Box, B) ∧ At(Bananas, C) ∧ Height(Monkey, Low) ∧ Height(Box, Low) ∧ Height(Banana, High) ∧ ¬Hold(Monkey, Banana).

Goal state:

Hold(Monkey, Banana)

(b) Q_2:

Action 1: GO(X, Y)

Precond At(Monkey, X) ∧ ¬ Height(~~Mo~~ Monkey, High) ∧ (X != Y)

Effect At(Monkey, Y) ∧ ¬ At(Monkey, X).

Action 2: Push(Box, X, Y)

Precond: At(Box, X) ∧ At(Monkey, X) ∧ Height(Monkey, Low) ∧ (X ≠ Y)

Effect: At(Box, Y) ∧ At(Monkey, Y)

Action 3: ClimbUp(X)

Precond: At(Monkey, X) ∧ At(Box, X) ∧ Height(Monkey, Low).

Effect: At(Monkey, X) ∧ At(Box, X) ∧ Height(Monkey, High) ∧ ¬Height(Monkey, L

Action 4: ClimbDown(X)

Precond: At(Monkey, X) ∧ At(Box, X) ∧ Height(Monkey, High)

Effect: Height(Monkey, Low) ∧ ¬Height(Monkey, High) ∧ At(Monkey, X) ∧ At(Box, X).

Action 5. Grasp(X, h)

Precond: At(Monkey, X) ∧ At(Bananas, X) ∧ Height(Monkey, h) ∧ Height(Bananas,

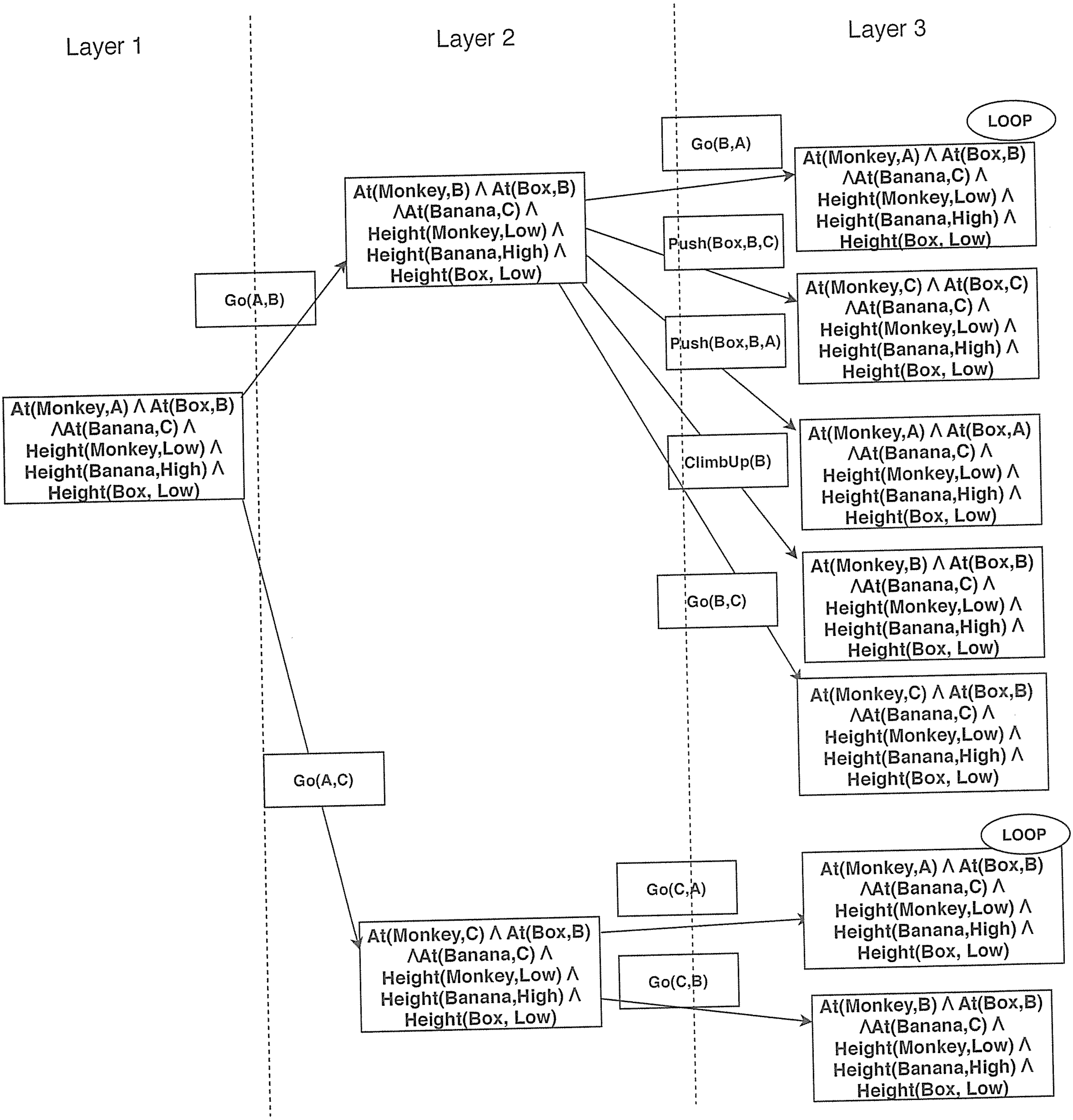
Effect: Holds(Monkey, Banana).

Action 6: UnGrasp(Bananas)

Precond: Holds(Monkey, Bananas)

Effect: ¬Holds(Monkey, Bananas).

Q3.

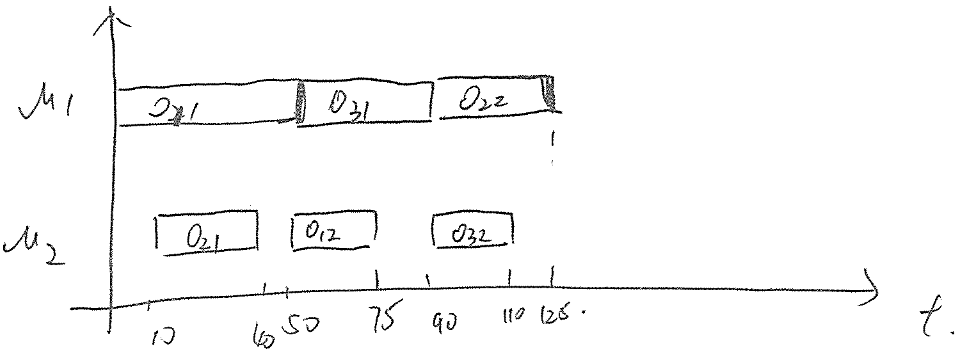

Layer 1
Layer 2
Layer 3
At(Monkey,A) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)
Go(A,B)
Go(A,C)
At(Monkey,B) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)
At(Monkey,C) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)
Go(B,A)
Push(Box,B,C)
Push(Box,B,A)
ClimbUp(B)
Go(B,C)
LOOP
At(Monkey,A) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)
At(Monkey,C) ∧ At(Box,C) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)
At(Monkey,A) ∧ At(Box,A) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)
At(Monkey,B) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)
At(Monkey,C) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)
Go(C,A)
Go(C,B)
LOOP
At(Monkey,A) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)
At(Monkey,B) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low)

Q4

Initial State	At(Monkey,A) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low) ∧¬Hold(Monkey, Banana)
Action 1	Go(A,B)
State 1	At(Monkey,B) ∧ At(Box,B) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low) ∧¬Hold(Monkey, Banana)
Action 2	Push(Box,B,C)
State 2	At(Monkey,C) ∧ At(Box,C) ∧At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Banana,High) ∧ Height(Box, Low) ∧¬Hold(Monkey, Banana)
Action 3	ClimbUp(C)
State 3	At(Monkey,C) ∧ At(Box,C) ∧At(Banana,C) ∧ Height(Monkey,High) ∧ Height(Banana,High) ∧ Height(Box, Low) ∧¬Hold(Monkey, Banana)
Action 4	Grasp(C,High)
State 4(Goal State)	Hold(Monkey, Banana) (∧At(Monkey,C) ∧ At(Box,C) ∧At(Banana,C) ∧ Height(Monkey,High) ∧ Height(Banana,High) ∧ Height(Box, Low))

Part 2.

$P(O_{11}, M_1, t_1) \rightarrow P(O_{21}, M_2, t_2) \rightarrow P(O_{31}, M_1, t_3) \rightarrow P(O_{12}, M_2, t_4)$
$\rightarrow P(O_{22}, M_1, t_5) \rightarrow P(O_{32}, M_2, t_6)$.



① Job Ready Time.

J_1	$O_{11} = 0$	$O_{12} = +\infty$
J_2	$O_{21} = +\infty$	$O_{22} = +\infty$
J_3	$O_{31} = +\infty$	$O_{32} = +\infty$

Machine Idle Time

$M_1 = 0$

$M_2 = 0$

Operation $(O_{11}, M_1, \text{Proc Time} = 0) = -50$ $\quad t_1 = 0$.

② J_1 $\cancel{O_{11}}$ = $O_{12} = 50$

J_2	$O_{21} = 10$	$O_{22} = +\infty$
J_3	$O_{31} = 20$	$O_{32} = +\infty$

MIT

$M_1 = 50$

$M_2 = 0$

operation $(O_{21}, M_2, t_2) \Rightarrow$ Operation $(O_{21}, M_2, \text{ProcTime} = 10) = -30$ $\quad t_2 = 10$.

③ JRT

J_1	$\cancel{O_{11}}$	$O_{12} = 50$
J_2	$\cancel{O_{21}}$	$O_{22} = 40$
J_3	$O_{31} = 20$	$O_{32} = +\infty$

MIT

$M_1 = 50$

$M_2 = 40$

(b)

Finishing time for $J_1 = t_4 + Proc(O_{12}) = 75$

Finishing time for $J_2 = t_5 + Proc(O_{22}) = 125$

Finishing time for $J_3 = t_6 + Proc(O_{32}) = 110$.

The makespan of this solution is the time of the job finished latest which is 125.

(c):

~~step 1.~~
~~earliest Idle Time (M_1) = 0, earliest Idle Time (M_2) = 0.~~
~~earliest Ready Time (O_{11}) = 0~~

step 1

Partial solution : Process (O_{11}, M_1, 0)

earliest Idle Time (M_1) = 50, earliest Idle Time (M_2) = 0.

earliest Ready Time (O_{12}) = 50, ~~earliest Ready Time = 0,~~

earliest Ready Time (O_{21}) = 10, earliest Ready Time (O_{22}) = $+\infty$

earliest Ready Time (O_{31}) = 20, earliest Ready Time (O_{32}) = $+\infty$

step 2

Partial solution : Process (O_{11}, M_1, 0) → Process(O_{21}, M_2, 10)

earliest Idle Time (M_1) = 50, earliest Idle Time (M_2) = 40

earliest Ready Time (O_{12}) = 50

earliest Ready Time (O_{22}) = 40

earliest Ready Time (O_{31}) = 20 earliest Ready Time (O_{32}) = $+\infty$

step 3.

Partial solution : Process (O_{11}, M_1, 0) → Process (O_{21}, M_2, 10) → Process(O_{12}, M_2, 50, earliest Idle Time (M_1) = 50, earliest Idle Time (M_2) = 25.

earliest Ready Time (O_{22}) = 40

earliest Ready Time (O_{31}) = 20 earliest Ready Time (O_{32}) = t20

Step 4.

Partial solution: Process (O_{11}, M_1, 0) → Process (O_{21}, M_2, 10)

⋮

Final solution

Process (O_{11}, M_1, 0) → Process (O_{21}, M_2, 10) → Process (O_{12}, M_2, 50)

→ Process (O_{22}, M_1, 50) → Process (O_{31}, M_1, 85) → Process (O_{32}, M_2, 125)

4): (SPT Rule)

Finishing time for Job 1. = 50 + Process Time (O_{12}) = 75.

Finishing time for Job 2 = 50 + Process Time (O_{22}) = 85.

Finishing time for Job 3 = 125 + Process Time (O_{32}) = 145.

The makespan of the solution using SPT rule is 145.

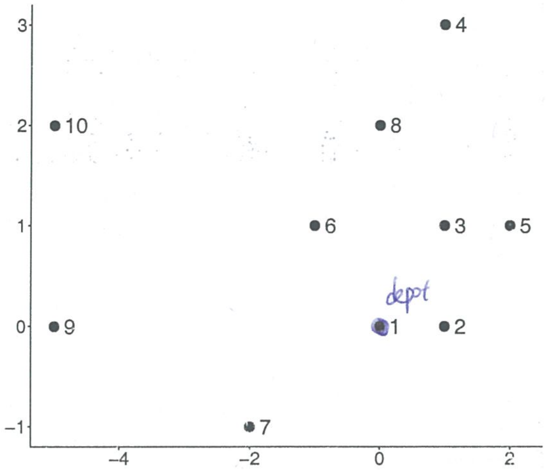
Comparing the makespans of using SPT and FCFS, the makespan of the solution using FCFS performs better, it has smaller makespan.

Part 3: Vehicle Routing [25 marks]

In this part, you are required to find a solution using the nearest neighbor heuristic and calculate its total cost for the given vehicle routing problem.

Problem Description

The graph below gives a vehicle routing problem.



The location and demand of each node is given as follows. Node 1 is the depot. Each node except the depot has a demand of 1. The capacity is 3.

Node	x-coordination	y-coordination	Demand
1 (depot)	0	0	0
2	1	0	1
3	1	1	1
4	1	3	1
5	2	1	1
6	-1	1	1
7	-2	-1	1
8	0	2	1
9	-5	0	1
10	-5	2	1

The cost of each edge is the Euclidean distance between the two end-nodes. The Euclidean distance matrix is given below.

$$
D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\
\begin{pmatrix}
0 & 1.00 & 1.41 & 3.16 & 2.24 & 1.41 & 2.23 & 2.00 & 5.00 & 5.39 \\
1.00 & 0 & 1.00 & 3.00 & 1.41 & 2.24 & 3.16 & 2.24 & 6.00 & 6.32 \\
1.41 & 1.00 & 0 & 2.00 & 1.00 & 2.00 & 3.61 & 1.41 & 6.08 & 6.08 \\
3.16 & 3.00 & 2.00 & 0 & 2.24 & 2.83 & 5.00 & 1.41 & 6.71 & 6.08 \\
2.24 & 1.41 & 1.00 & 2.24 & 0 & 3.00 & 4.47 & 2.24 & 7.07 & 7.07 \\
1.41 & 2.24 & 2.00 & 2.83 & 3.00 & 0 & 2.24 & 1.41 & 4.12 & 4.12 \\
2.23 & 3.16 & 3.61 & 5.00 & 4.47 & 2.24 & 0 & 3.61 & 3.16 & 4.24 \\
2.00 & 2.24 & 1.41 & 1.41 & 2.24 & 1.41 & 3.61 & 0 & 5.39 & 5.00 \\
5.00 & 6.00 & 6.08 & 6.71 & 7.07 & 4.12 & 3.16 & 5.39 & 0 & 2.00 \\
5.39 & 6.32 & 6.08 & 6.08 & 7.07 & 4.12 & 4.24 & 5.00 & 2.00 & 0
\end{pmatrix}
\end{array}
$$

8

5). No, it doesn't mean FCFS rule can always generates better solution than using SPT rule. We can only say FCFS rule is more suitable to apply on this particular problem.

Part 3.

1): $R_1 = (1, 2, 3\ 5, 1)$

$R_2 = (1, 6, 8, 4, 1)$

$R_3 = (1, 7, 9, 10, 1)$

2):

$D_1 = 1 + 1 + 1 + 2.24 = 5.24$

$D_2 = 1.41 + 1.41 + 1.41 + 3.16 = 7.39$

$D_3 = 2.23 + 3.16 + 2.00 + 5.39 = 12.78$

Total Distance $= D_1 + D_2 + D_3 = 25.41$

3): Function set: ① Multiply

② Add

③ Sub

④ Protected Division

⑤. Square root

Terminal set : ① Capacity of truck

② Demand of each node

③. Distance to Depot ~~s~~ storage

④ Distance to the ~~neighbo~~ nearest neighbour

⑤ current ~~loads on~~ number of depot on the truck

The Fitness Function we can use the total distance, therefore the smaller result we get the better ~~solution~~ it performs.

For the function set, I want to use Arithmetic operators and square root. Because in this problems, we don't have lots features, and this is a relatively easy problem, hence I think we don't ~~to~~ need generate a GP Tree too complex.

~~For the terminal set, w we can set the ca~~

Terminal set:

For this problem we should consider the current number of depot on the truck, if we only have the last one on the truck we should not go too far away ~~fr~~ from the depot station, ~~which could~~ ~~Also~~

~~we~~ Hence the distance from current node to depot station is also involved. We also want to visit the nearest neighbour so the terminal set also includes the distance to the nearest neighbour.

For this particular problem we ~~don't~~ do not need to care too much about the capacity of truck and the Demand of each node, but to make the code more general, I added these two in to the ~~function~~ terminal set.

Fitness Function

In this ~~path~~ problem, we aim to find the shortest path, so we can evaluate the heuristic based on the ~~total~~ total ~~to~~ distance of the solution generated by the heuristic.