Problem Solving
Master 1, Visual Computing
2024/2025

Exam

Exercise 1:

Part 1: (4 points)

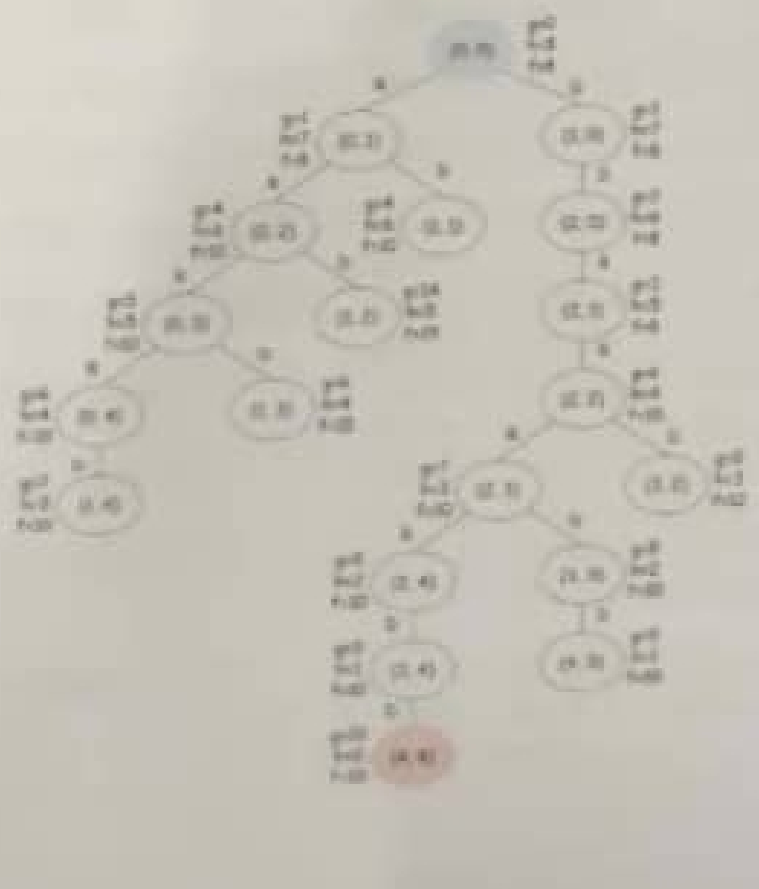
State: (x, y) => Node [(x, y), g, h] (x: vertical coordinate, y: horizontal coordinate)
Initial state: (0, 0) => Initial node [(0, 0), 0, 8]
Goal state: (4, 4)
Actions: R (right), L (left), D (down), U (up)
Path Cost (g): The path cost g(n) is the accumulated movement cost from the **start state** to the current state. For example:

- Moving through **Plain (P)** cells: g increases by 1 per step.
- Moving through **Forest (F)** cells: g increases by 3 per step.
- Moving through **Enemy Zone (E)** cells: g increases by 10 per step.

Heuristic Function (h): The heuristic estimates the cost from the current state to the goal state. We use the Manhattan Distance.
Objective: Find a path from (0, 0) to (4, 4) that minimizes the total cost f(n), where: f(n)=g(n)+h(n).

OPEN	CLOSED
[(0,0),0,8]	
[(0,1),1,7], [(1,0),1,7]	[(0,0),0,8]
[(1,0),1,7], [(0,2),4,6], [(1,1),4,6]	[(0,0),0,8], [(0,1),1,7]
[(0,2),4,6], [(1,1),4,6], [(2,0),2,6]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7]
[(0,2),4,6], [(1,1),4,6], [(2,1),3,5]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6]
[(0,2),4,6], [(1,1),4,6], [(2,2),6,4]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5]
[(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(1,2),14,5],	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6]
[(2,2),6,4], [(0,3),5,5], [(1,2),14,5]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6]
[(0,3),5,5], [(1,2),14,5], [(2,3),7,3], [(3,2),9,3]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4]
[(1,2),14,5], [(2,3),7,3], [(3,2),9,3], [(0,4),6,4], [(1,3),6,4]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(2,1),5,5]
[(1,2),14,5], [(3,2),9,3], [(0,4),6,4], [(1,3),6,4], [(2,4),8,2], [(3,3),8,2]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(2,1),5,5], [(2,3),7,3]
[(1,2),14,5], [(3,2),9,3], [(1,3),6,4], [(2,4),8,2], [(3,3),8,2], [(1,4),7,3]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(2,1),5,5], [(2,3),7,3], [(0,4),6,4]

[(1,2),14,5], [(3,2),9,3], [(2,4),8,2], [(3,3),8,2], [(1,4),7,3]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(2,1),5,5], [(2,3),7,3], [(0,4),6,4], [(1,3),6,4]
[(1,2),14,5], [(3,2),9,3], [(3,3),8,2], [(1,4),7,3], [(3,4),9,1]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(2,1),5,5], [(2,3),7,3], [(0,4),6,8], [(1,3),6,4], [(2,4),8,2]
[(1,2),14,5], [(3,2),9,3], [(1,4),7,3], [(3,4),9,1], [(4,3),9,1]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(2,1),5,5], [(2,3),7,3], [(0,4),6,4], [(1,3),6,4], [(2,4),8,2], [(3,3),8,2]
[(1,2),14,5], [(3,2),9,3], [(3,4),9,1], [(4,3),9,1]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(2,1),5,5], [(2,3),7,3], [(0,4),6,4], [(1,3),6,4], [(2,4),8,2], [(3,3),8,2], [(1,4),7,3]
[(1,2),14,5], [(3,2),9,3], [(4,3),9,1], [(4,4),10,0]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(2,1),5,5], [(2,3),7,3], [(0,4),6,4], [(1,3),6,4], [(2,4),8,2], [(3,3),8,2], [(1,4),7,3], [(3,4),9,1]
[(1,2),14,5], [(3,2),9,3], [(4,4),10,0]	[(0,0),0,8], [(0,1),1,7], [(1,0),1,7], [(2,0),2,6], [(2,1),3,5], [(0,2),4,6], [(1,1),4,6], [(2,2),6,4], [(0,3),5,5], [(2,1),5,5], [(2,3),7,3], [(0,4),6,4], [(1,3),6,4], [(2,4),8,2], [(3,3),8,2], [(1,4),7,3], [(3,4),9,1], [(4,3),9,1]



Path:

	0	1	2	3	4
0	P	P	F	P	P
1	P	F	E	P	P
2	P	P	F	P	P
3	M	M	F	P	P
4	P	P	P	P	P

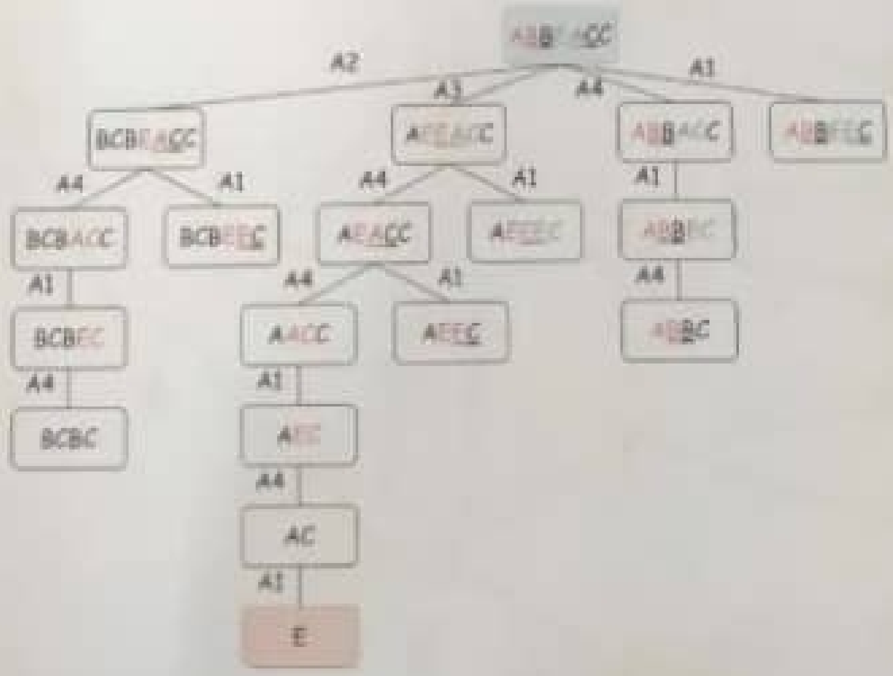
Cost: 10
Solution: D -> D -> D -> R -> R -> R -> R -> D -> D

Part 2: (4 points)

State: Sequence of characters
Initial state: ABBEACC
Goal state: E
Actions: A1, A2, A3, A4

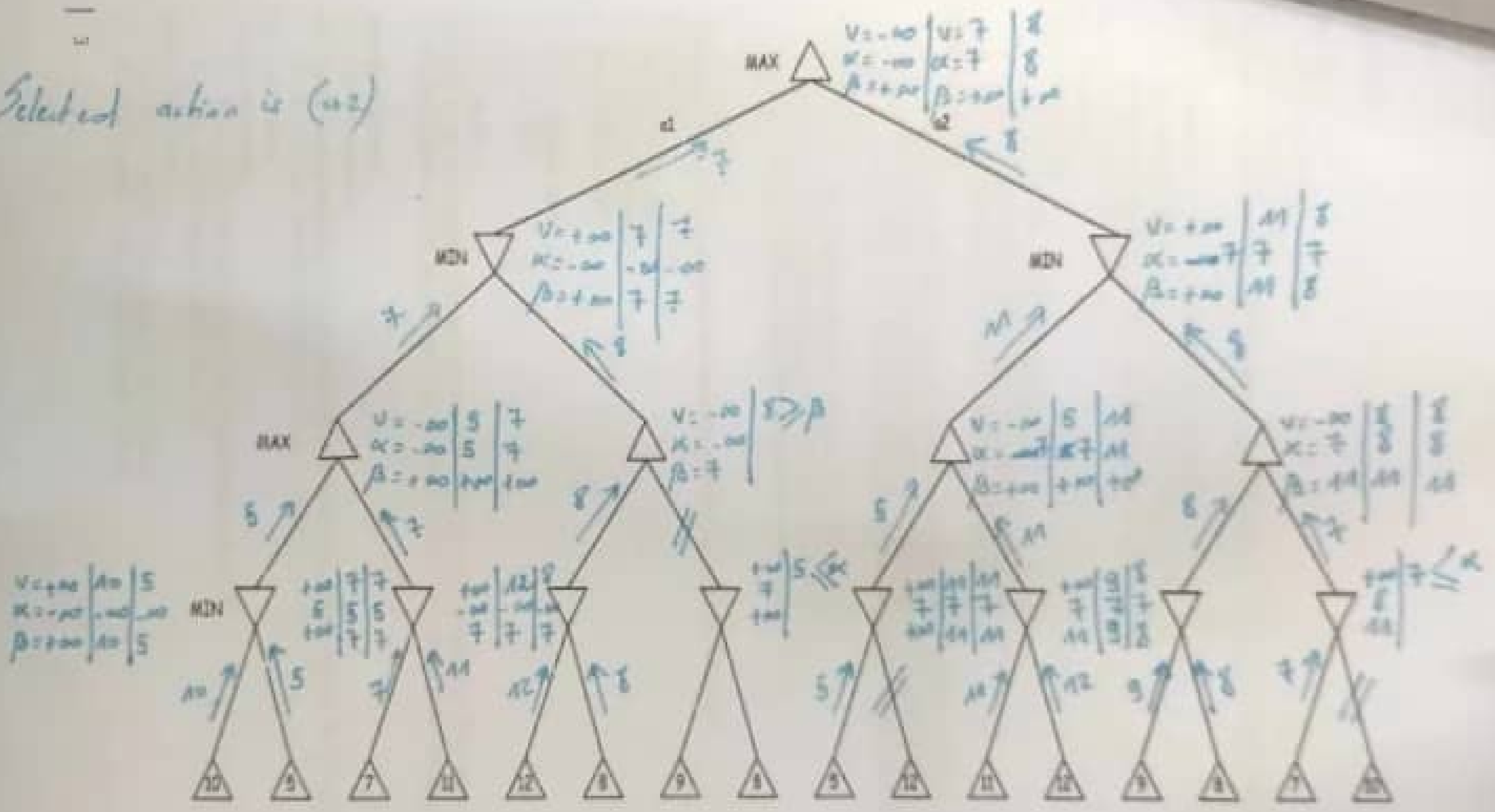
OPEN	CLOSED
ABBEACC	
BCBEACC, AEEACC, ABBACC, ABBEEC	ABBEACC
AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC	ABBEACC, BCBEACC
ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC	ABBEACC, BCBEACC, AEEACC
ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC, ABBEC	ABBEACC, BCBEACC, AEEACC, ABBACC
BCBACC, BCBEEC, AEACC, AEEEC, ABBEC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC
BCBEEC, AEACC, AEEEC, ABBEC, BCBEC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC
AEACC, AEEEC, ABBEC, BCBEC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC
AEEEC, ABBEC, BCBEC, AACC, AEEC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC
ABBEC, BCBEC, AACC, AEEC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC
BCBEC, AACC, AEEC, ABBC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC, ABBEC

AACC, AEEC, ABBC, BCBC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC, ABBEC, BCBEC
AEEC, ABBC, BCBC, AEC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC, ABBEC, BCBEC, AACC
ABBC, BCBC, AEC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC, ABBEC, BCBEC, AACC, AEEC
BCBC, AEC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC, ABBEC, BCBEC, AACC, AEEC, ABBC
AC	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC, ABBEC, BCBEC, AACC, AEEC, ABBC, BCBC
E	ABBEACC, BCBEACC, AEEACC, ABBACC, ABBEEC, BCBACC, BCBEEC, AEACC, AEEEC, ABBEC, BCBEC, AACC, AEEC, ABBC, BCBC, AC



Path: ABBEACC -> AEEACC -> AEACC -> AACC -> AEC -> AC -> E
Solution: A3 -> A4 -> A4 -> A1 -> A4 -> A1

Selected action is (a2)



Exercise 3:

Part 1: (4 points)

1. **CSP formulation**

a) **Variables and domains**

We represent the gardens from left (Garden 1) to right (Garden 3), each with 3 variables F_i, A_i and I_i, $i \in \{1,2,3\}$

- **Flowers** (F_1, F_2, F_3), with domain {daisies, tulips, roses}
- **Ages of Gardeners** (A_1, A_2, A_3), with domain {20, 30, 40}
- **Irrigation Systems** (I_1, I_2, I_3), with domain {hose, sprinkler, drip}

So, we have 9 variables in total.

b) **Constraints**

Unary Constraints:

1. F_1=daisies (Garden on the left has daisies).
2. I_2=drip (Garden in the middle is watered by a drip system)
3. A_2=30 (Middle garden is tended by a 30-year-old gardener)

Other Constraints:

4. $\exists i$, F_i=roses $\wedge$ I_i=sprinkler (The garden with roses is watered by a sprinkler system)
5. $\exists i$, A_i=20 $\wedge$ F_i=tulips (The youngest gardener, age 20, tends the garden with tulips).
6. $\exists i$, A_i=40 $\wedge$ (I_{i-1}=hose $\vee$ I_{i+1}=hose) (The oldest gardener, age 40, tends a garden next to one watered with a hose).
7. All gardens have **different flowers**: $(F_1 \neq F_2) \wedge (F_1 \neq F_3) \wedge (F_2 \neq F_3)$
8. All gardeners have **different ages**: $(A_1 \neq A_2) \wedge (A_1 \neq A_3) \wedge (A_2 \neq A_3)$.
9. All gardens have **different irrigation systems**: $(I_1 \neq I_2) \wedge (I_1 \neq I_3) \wedge (I_2 \neq I_3)$

2. **Arc consistency**

After applying the unary constraints, the variables will have the following domains:

F_1={daisies}
F_2={daisies, roses, tulips}
F_3={daisies, roses, tulips}
I_1={hose, sprinkler, drip}
I_2={drip}
I_3={hose, sprinkler, drip}
A_1={20, 30, 40}
A_2={30}
A_3={20, 30, 40}

The CSP is not arc-consistent, because for example constraint 6 is not satisfied. If A_i=40, i=1 or i=3. If A_1=40, then I_2 must be hose, but I_2=drip. If A_3=40, then I_2 must be hose, which is not the case.

Part 2: (4 points)

1. **CSP formulation**

a) **Variables and domains:** Each coworker W_i is assigned to one of the offices. W={W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8}. Domain (W_i)={B_1, B_2, B_3, B_4}, for i=1, 2, ..., 8.

b) **Constraints**

Unary constraints :

1. W_1, W_3, W_5 $\in$ {B_1, B_2} (W_1, W_3, and W_5 require Offices B_1 or B_2).
2. W_8 $\in$ {B_3, B_4} (W_8 require Offices B_3 or B_4).

Other Constraints:

3. $W_1 \neq W_2$ (W_1 and W_2 cannot share the same office).
4. $W_3 \neq W_4$ (W_3 and W_4 cannot share the same office).

5. $W_7 \neq W_8$ (W_7 and W_8 cannot share the same office).
6. $W_5=W_6$ (W_5 and W_6 must work in the same office).
7. $W_1=W_7$ (W_2 and W_7 must work in the same office).
8. $\text{Count}(W_i=B_1)\leq 3$ (Office B_1 has a maximum capacity of 3).
9. $\text{Count}(W_i=B_2)\leq 2$ (Office B_2 has a maximum capacity of 2).
10. $\text{Count}(W_i=B_3)\leq 2$ (Office B_3 has a maximum capacity of 2).
11. $\text{Count}(W_i=B_4)\leq 3$ (Office B_4 has a maximum capacity of 3).

2. Solving the CSP

First, we apply the unary constraints and the similarity constraints to reduce the variables' domains. Then, we apply the Backtracking algorithm with forward checking and use MRV and Degree heuristics for variable selection.

Assignment	**Forward checking**	**Domains after forward checking**	**Remaining constraints**	**Comments**
{}		$W_1=\{B_1, B_2\}$ $W_2=\{B_1, B_2, B_3, B_4\}$ $W_3=\{B_1, B_2\}$ $W_4=\{B_1, B_2, B_3, B_4\}$ $W_5=\{B_1, B_2\}$ $W_6=\{B_1, B_2\}$ $W_7=\{B_1, B_2\}$ $W_8=\{B_3, B_4\}$	2 1 1 1 2 1	Select W_1
$\{W_1=B_1\}$	C_3 and C_7	$W_2=\{B_2, B_3, B_4\}$ $W_3=\{B_1, B_2\}$ $W_4=\{B_1, B_2, B_3, B_4\}$ $W_5=\{B_1, B_2\}$ $W_6=\{B_1, B_2\}$ $W_7=\{B_1\}$ $W_8=\{B_3, B_4\}$		Select W_7
$\{W_1=B_1, W_7=B_1\}$		$W_2=\{B_2, B_3, B_4\}$ $W_3=\{B_1, B_2\}$ $W_4=\{B_1, B_2, B_3, B_4\}$ $W_5=\{B_1, B_2\}$ $W_6=\{B_1, B_2\}$ $W_8=\{B_3, B_4\}$	 1 1 1 o	Select W_3
$\{W_1=B_1, W_7=B_1, W_3=B_1\}$	C_4 and C_8	$W_2=\{B_2, B_3, B_4\}$ $W_4=\{B_2, B_3, B_4\}$ $W_5=\{B_2\}$ $W_6=\{B_2\}$ $W_8=\{B_3, B_4\}$	 1 1 0	Select W_5
$\{W_1=B_1, W_7=B_1, W_3=B_1, W_5=B_2\}$	C_6	$W_2=\{B_2, B_3, B_4\}$ $W_4=\{B_2, B_3, B_4\}$ $W_6=\{B_2\}$ $W_8=\{B_3, B_4\}$		Select W_6
$\{W_1=B_1, W_7=B_1, W_3=B_1, W_5=B_2, W_6=B_2\}$	C_9	$W_2=\{B_3, B_4\}$ $W_4=\{B_3, B_4\}$ $W_8=\{B_3, B_4\}$		Select W_2
$\{W_1=B_1, W_7=B_1, W_3=B_1, W_5=B_2, W_6=B_2, W_2=B_3\}$		$W_4=\{B_3, B_4\}$ $W_8=\{B_3, B_4\}$		Select W_4

{W_1=B_1, W_7=B_1, W_3=B_1, W_5=B_2, W_6=B_2, W_2=B_3, W_4=B_3}	C_{10}	W_8={B_4}		Select W_8
{W_1=B_1, W_7=B_1, W_3=B_1, W_5=B_2, W_6=B_2, W_2=B_3, W_4=B_3, W_8=B_4}				