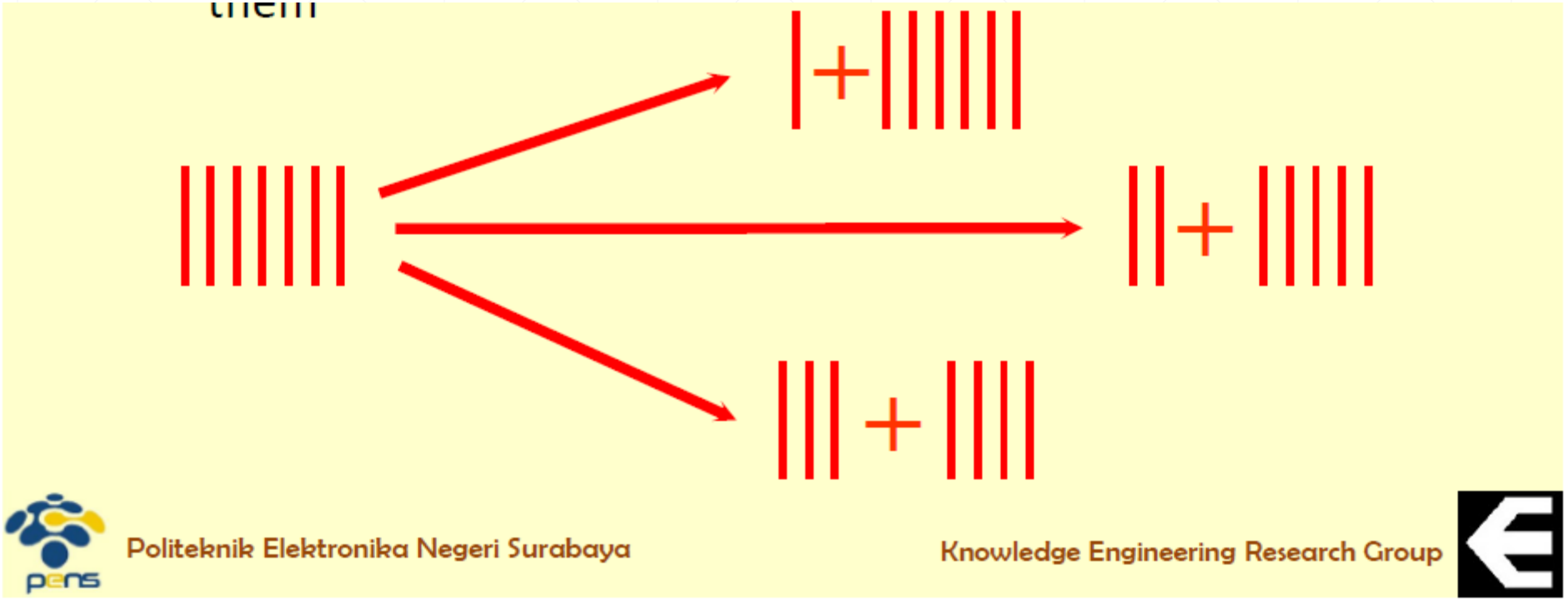
Reinforcement Learning – NIM Game

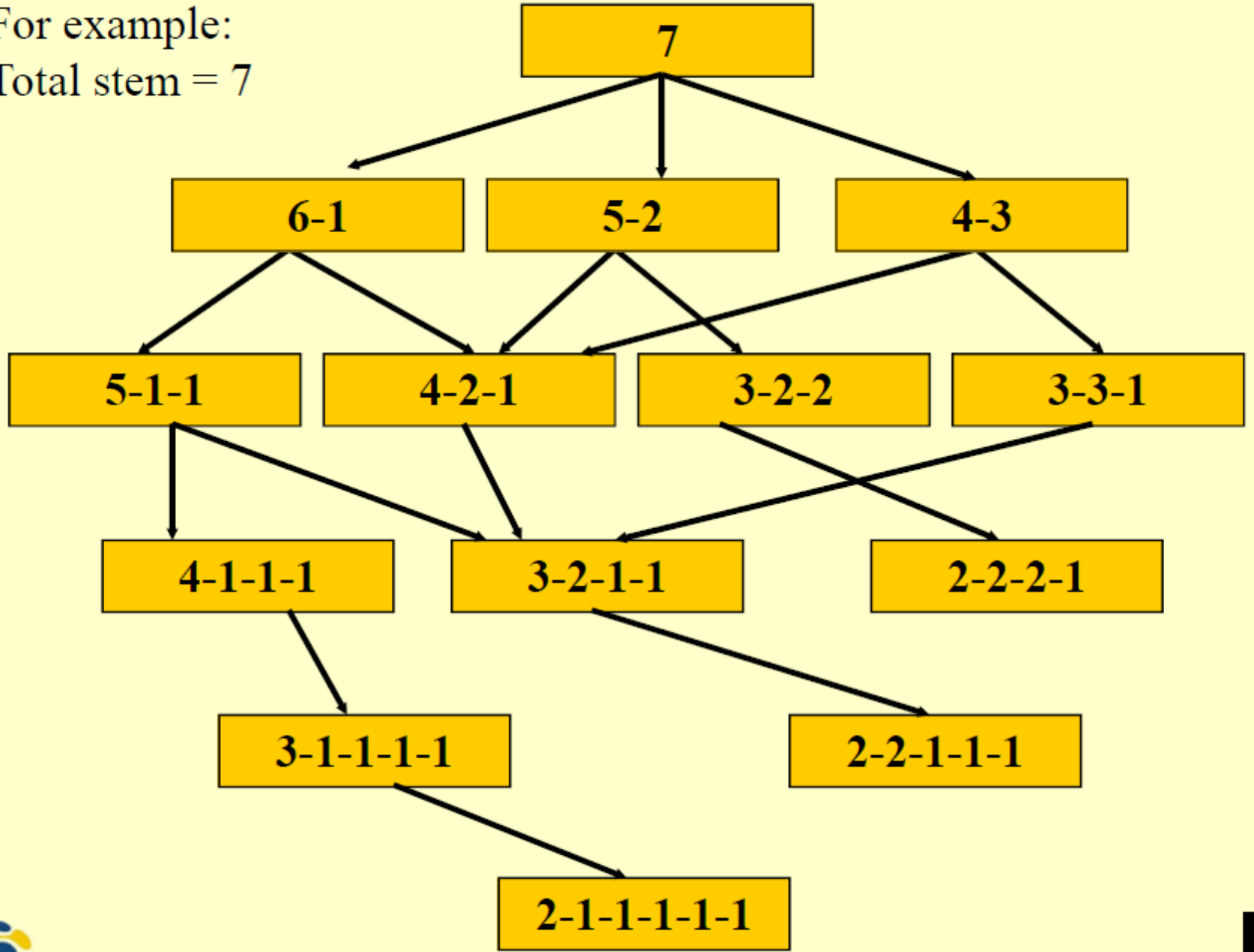
Hanif Izzudin Rahman
S2 Teknik Elektro
PENS 2020

NIM GAME

- Series of stems
- Each player has to divide one of item groups into 2 groups those have not different number of members for the group, and have not empty number of members for one of them



For example:
Total stem = 7



Value function

- Uses Temporal Difference Learning

  $V(s) = V(s) + \alpha [V'(s) - V(s)$

- Initialization
  - Computer wins : $V(s) \leftarrow 1$
  - Computer loses : $V(s) \leftarrow 0$
  - Other states: $V(s) \leftarrow 0.5$
  - Learning rate $\rightarrow \alpha = 0.1$

Exploratory move

- Greedy movement
  - Select highest V’(s)
- ε-greedy movement
  - Tends to select highest V’(s)
  - But there is possibility to select V’(s) randomly

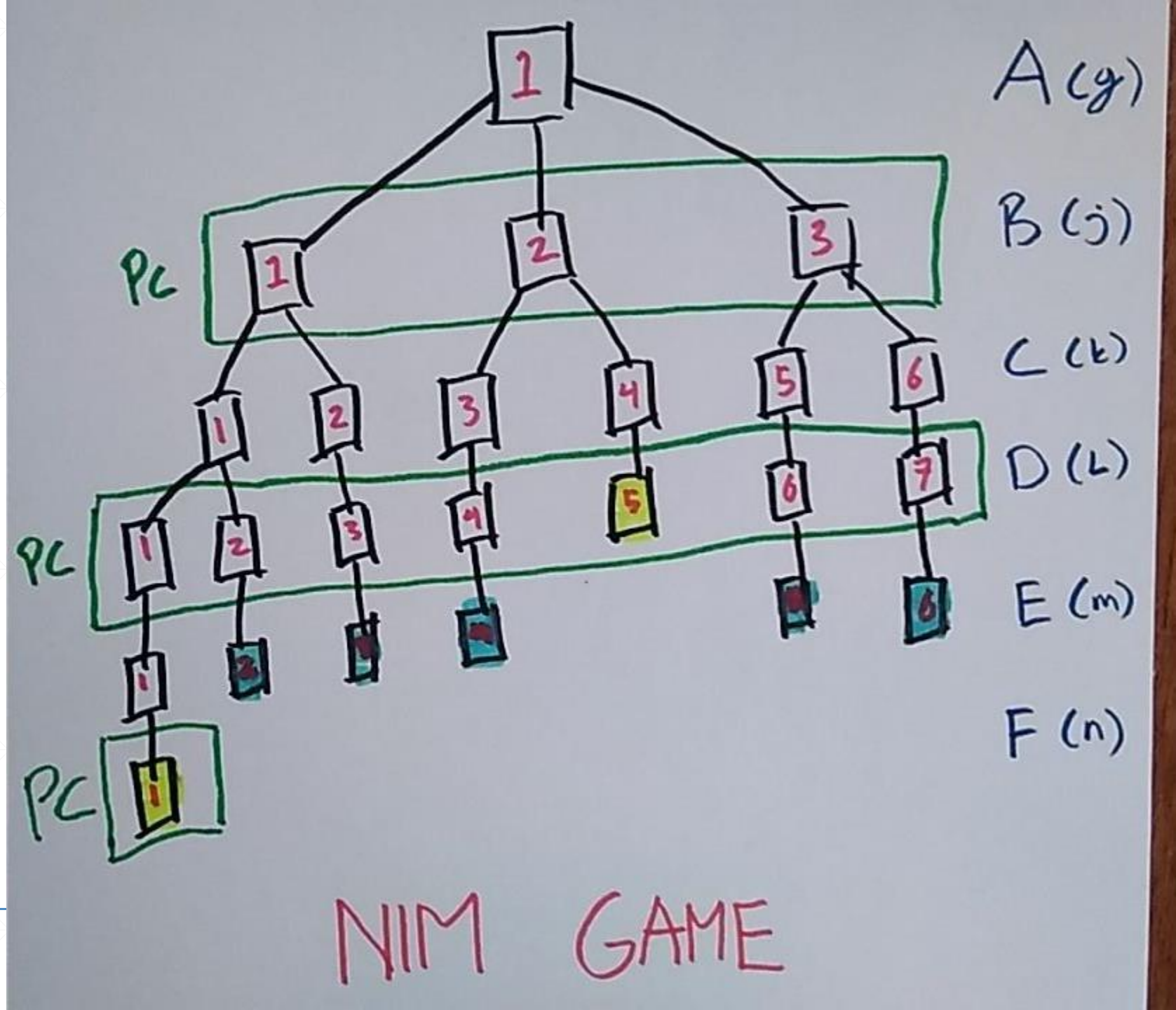
```
p ←   random[0..1]
if (p<0.1) {
   selected child ←   select one of children randomly
} else {
   selected child ←   select one of children those has highest V’(s)
}
```

Program

https://github.com/hanifizzudinrahman/Reinforcment-Learning-NIM-Game-Python-Basic-

Task 1 – NIM Game

(“Computer” First Start The Game)



- Initialization

```
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.  0.5 0.5 0.5 0.5 0.5 0.5]
D =  [0.  0.5 0.5 0.5 0.5 1.  0.5 0.5]
E =  [0.  0.5 0.  0.  0.  0.  0. ]
F =  [0. 1.]
```

- Reinforcement Learning

```
1 Step-by-step => 1 2 3 4 4 0
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.  0.5 0.5 0.5 0.5 0.5 0.5]
D =  [0.   0.5  0.5  0.5  0.45 1.   0.5  0.5 ]
E =  [0.  0.5 0.  0.  0.  0.  0. ]
F =  [0. 1.]

2 Step-by-step => 1 3 5 6 5 0
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.  0.5 0.5 0.5 0.5 0.5 0.5]
D =  [0.   0.5  0.5  0.5  0.45 1.   0.45 0.5 ]
E =  [0.  0.5 0.  0.  0.  0.  0. ]
F =  [0. 1.]


3 Step-by-step => 1 1 2 3 3 0
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.  0.5 0.5 0.5 0.5 0.5 0.5]
D =  [0.   0.5  0.5  0.45 0.45 1.   0.45 0.5 ]
E =  [0.  0.5 0.  0.  0.  0.  0. ]
F =  [0. 1.]

4 Step-by-step => 1 2 4 5 0 0
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.   0.5  0.5  0.5  0.55 0.5  0.5 ]
D =  [0.   0.5  0.5  0.45 0.45 1.   0.45 0.5 ]
E =  [0.  0.5 0.  0.  0.  0.  0. ]
F =  [0. 1.]
```

- Testing

Iteration = 10

```
==> WIN By Stupid ENEMY <==
1 Step-by-step => 1 2 4 5 0 0

==> LOSE by Smart ENEMY <==
2 Step-by-step => 1 2 3 4 4 0

==> WIN By Stupid ENEMY <==
3 Step-by-step => 1 2 4 5 0 0

==> WIN By Stupid ENEMY <==
4 Step-by-step => 1 2 4 5 0 0

==> LOSE by Smart ENEMY <==
5 Step-by-step => 1 2 3 4 4 0

==> LOSE by Smart ENEMY <==
6 Step-by-step => 1 2 3 4 4 0
```

Iteration = 100

```
==> LOSE by Smart ENEMY <==
1 Step-by-step => 1 2 3 4 4 0

==> WIN By Stupid ENEMY <==
2 Step-by-step => 1 2 4 5 0 0

==> WIN By Stupid ENEMY <==
3 Step-by-step => 1 2 4 5 0 0

==> LOSE by Smart ENEMY <==
4 Step-by-step => 1 2 3 4 4 0

==> WIN By Stupid ENEMY <==
5 Step-by-step => 1 2 4 5 0 0

==> LOSE by Smart ENEMY <==
6 Step-by-step => 1 2 3 4 4 0
```

Iteration = 1000

```
==> WIN By Stupid ENEMY <==
1 Step-by-step => 1 2 4 5 0 0

==> WIN By Stupid ENEMY <==
2 Step-by-step => 1 2 4 5 0 0

==> WIN By Stupid ENEMY <==
3 Step-by-step => 1 2 4 5 0 0

==> WIN By Stupid ENEMY <==
4 Step-by-step => 1 2 4 5 0 0

==> LOSE by Smart ENEMY <==
5 Step-by-step => 1 2 3 4 4 0

==> WIN By Stupid ENEMY <==
6 Step-by-step => 1 2 4 5 0 0
```

Iteration = 10000

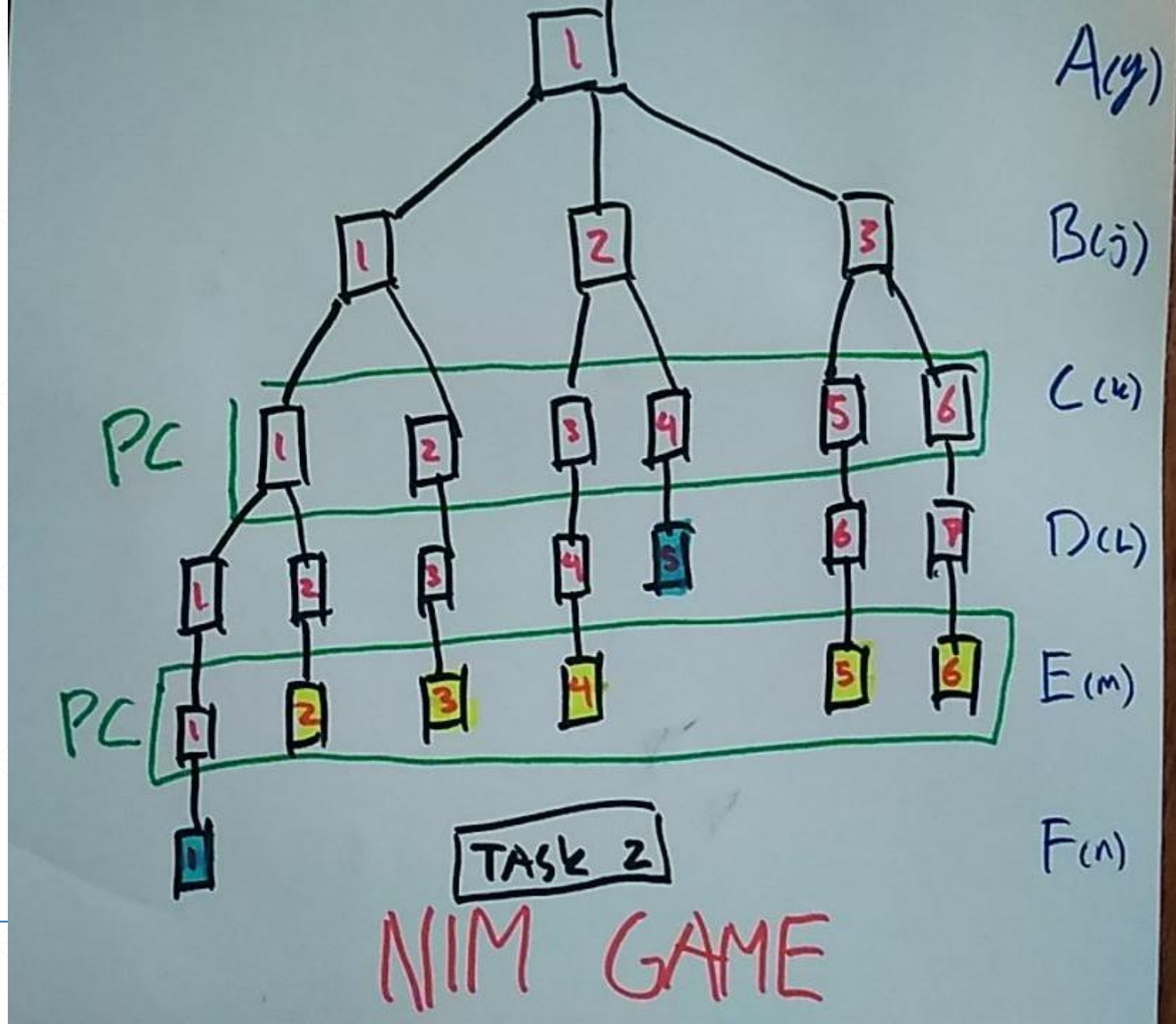
```
==> LOSE by Smart ENEMY <==
1 Step-by-step => 1 2 3 4 4 0

==> LOSE by Smart ENEMY <==
2 Step-by-step => 1 2 3 4 4 0

==> LOSE by Smart ENEMY <==
3 Step-by-step => 1 2 3 4 4 0

==> WIN By Stupid ENEMY <==
4 Step-by-step => 1 2 4 5 0 0

==> LOSE by Smart ENEMY <==
5 Step-by-step => 1 2 3 4 4 0

==> LOSE by Smart ENEMY <==
6 Step-by-step => 1 2 3 4 4 0
```

Sometimes **Win**

Sometimes **Lose**

Task 2 – NIM Game

(“Enemy” First Start The Game)



- Initialization

```
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.  0.5 0.5 0.5 0.5 0.5 0.5]
D =  [0.  0.5 0.5 0.5 0.5 0.  0.5 0.5]
E =  [0.  0.5 1.  1.  1.  1.  1. ]
F =  [0. 0.]
```

- Reinforcement Learning

```
1 Step-by-step => 1 2 3 4 4 0
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.  0.5 0.5 0.5 0.5 0.5 0.5]
D =  [0.   0.5  0.5  0.5  0.55 0.   0.5  0.5 ]
E =  [0.  0.5 1.  1.  1.  1.  1. ]
F =  [0. 0.]

2 Step-by-step => 1 2 3 4 4 0
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.    0.5   0.5   0.505 0.5   0.5   0.5  ]
D =  [0.    0.5   0.5   0.5   0.595 0.    0.5   0.5  ]
E =  [0.  0.5 1.  1.  1.  1.  1. ]
F =  [0. 0.]

3 Step-by-step => 1 1 2 3 3 0
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.    0.5   0.5   0.505 0.5   0.5   0.5  ]
D =  [0.    0.5   0.5   0.55  0.595 0.    0.5   0.5  ]
E =  [0.  0.5 1.  1.  1.  1.  1. ]
F =  [0. 0.]

4 Step-by-step => 1 3 5 6 5 0
A =  [0.  0.5]
B =  [0.  0.5 0.5 0.5]
C =  [0.    0.5   0.5   0.505 0.5   0.5   0.5  ]
D =  [0.    0.5   0.5   0.55  0.595 0.    0.55  0.5  ]
E =  [0.  0.5 1.  1.  1.  1.  1. ]
F =  [0. 0.]
```

- Testing

Iteration = 10	Iteration = 100	Iteration = 1000	Iteration = 10000
==> WIN SMART <== 1 Step-by-step => 1 2 3 4 4 0	==> WIN SMART <== 9 Step-by-step => 1 2 3 4 4 0	==> WIN By Stupid ENEMY <== 1 Step-by-step => 1 3 5 6 5 0	==> WIN SMART <== 1 Step-by-step => 1 2 3 4 4 0
==> WIN By Stupid ENEMY <== 2 Step-by-step => 1 1 1 2 2 0	==> WIN SMART <== 10 Step-by-step => 1 2 3 4 4 0	==> WIN SMART <== 2 Step-by-step => 1 1 2 3 3 0	==> WIN SMART <== 2 Step-by-step => 1 1 2 3 3 0
==> WIN By Stupid ENEMY <== 3 Step-by-step => 1 3 5 6 5 0	==> LOSE by Smart ENEMY <== 11 Step-by-step => 1 1 1 1 1 1	==> WIN SMART <== 3 Step-by-step => 1 2 3 4 4 0	==> WIN By Stupid ENEMY <== 3 Step-by-step => 1 3 5 6 5 0
==> WIN By Stupid ENEMY <== 4 Step-by-step => 1 3 5 6 5 0	==> WIN By Stupid ENEMY <== 12 Step-by-step => 1 3 6 7 6 0	==> WIN By Stupid ENEMY <== 4 Step-by-step => 1 3 5 6 5 0	==> WIN SMART <== 4 Step-by-step => 1 2 3 4 4 0
==> WIN By Stupid ENEMY <== 5 Step-by-step => 1 3 5 6 5 0	==> WIN By Stupid ENEMY <== 13 Step-by-step => 1 3 6 7 6 0	==> WIN SMART <== 5 Step-by-step => 1 2 3 4 4 0	==> WIN By Stupid ENEMY <== 5 Step-by-step => 1 3 5 6 5 0
==> LOSE by Smart ENEMY <== 6 Step-by-step => 1 1 1 1 1 1	==> WIN SMART <== 14 Step-by-step => 1 2 3 4 4 0	==> WIN By Stupid ENEMY <== 6 Step-by-step => 1 3 5 6 5 0	==> WIN SMART <== 6 Step-by-step => 1 1 2 3 3 0

Sometimes **Win**

Sometimes **Lose**

Always **Win**