Fundamentals of Artificial Intelligence

Laboratory

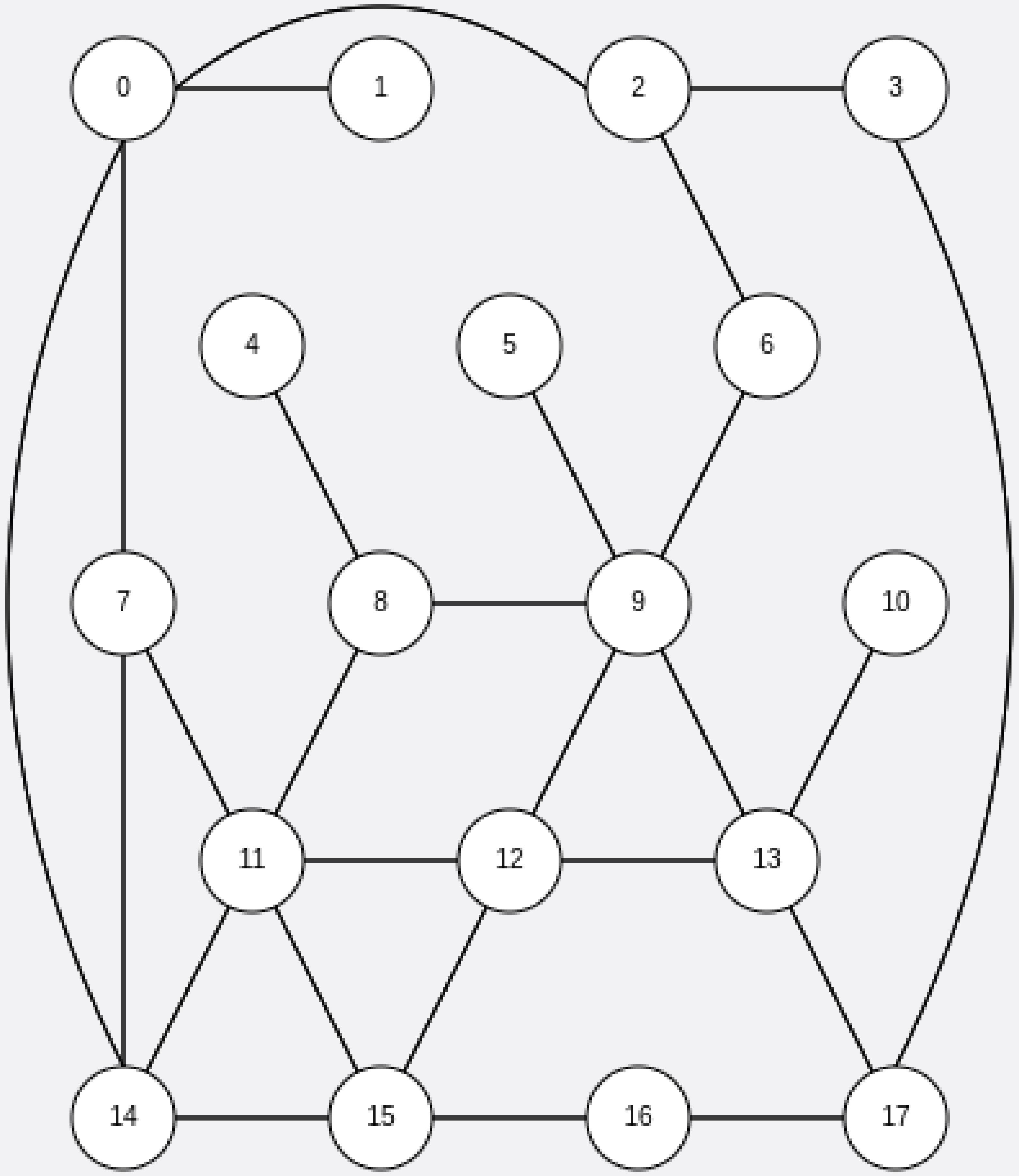
Dr. Mauro Dragoni

Department of Information Engineering and Computer Science

Academic Year 2021/2022

Exercise 3.18

- Apply both **BFS** and **DFS** for exploring the **entire graph**.
  Start node: **0**. Exploration in descending order. Report the list of the explored nodes.



Exercise 3.18 - Solution

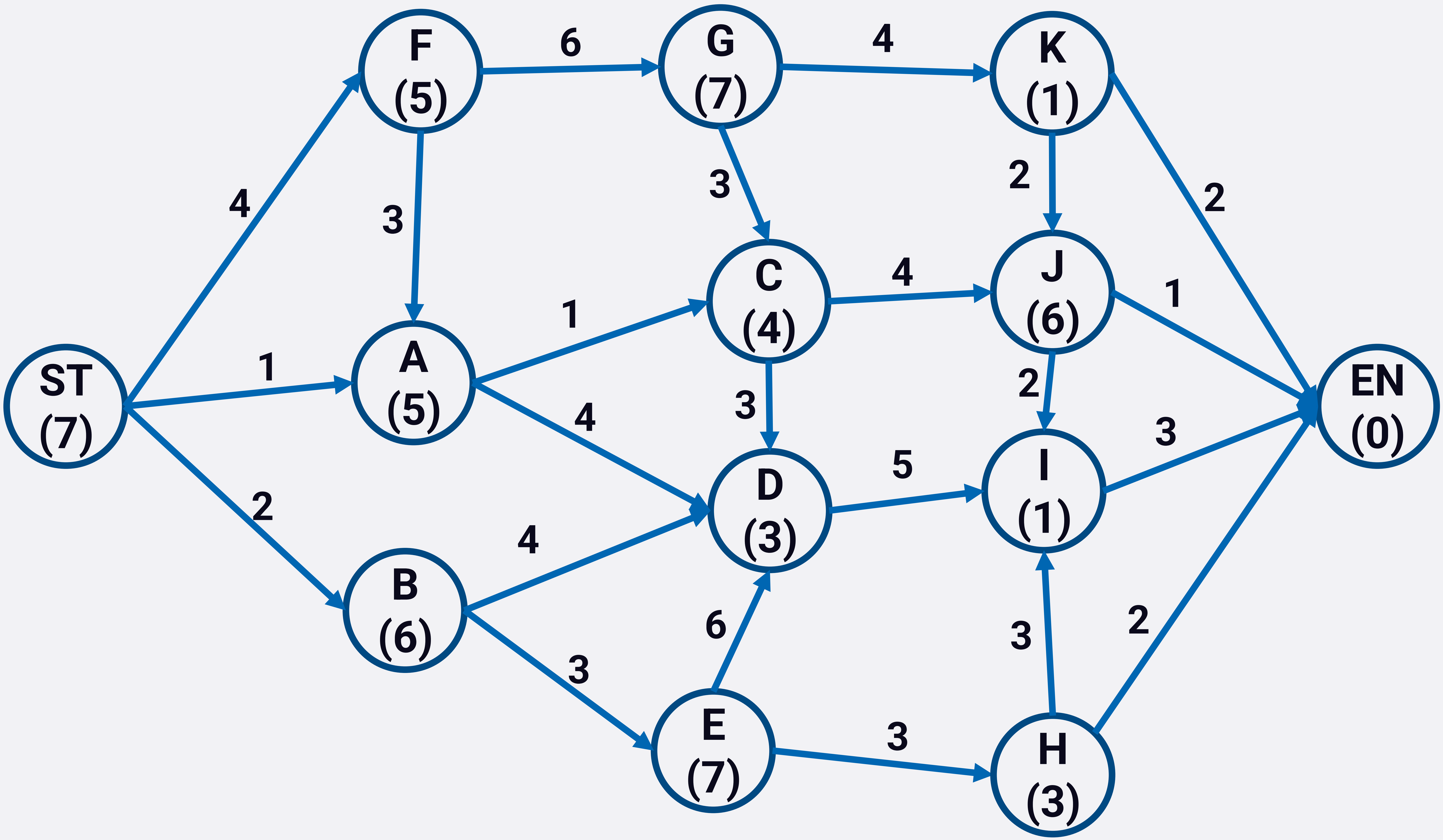
- **BFS**

  S0 = {0}
  S1 = {14,7,2,1}
  S2 = {15,11,6,3}
  S3 = {16,12,8,9,17}
  S4 = {13,4,5}
  S5 = {10}

- **DFS**

  S0 = {0,14,15,16,17,13,12,11,8,9,6,2,3} → Backtrack to 9
  S1 = {5} → Backtrack to 8
  S2 = {4} → Backtrack to 11
  S3 = {7} → Backtrack to 13
  S4 = {10} → Backtrack to 0
  S5 = {1}

Exercise 3.19



Exercise 3.19

Step	Queue	Processed Nodes	Children
1	ST(7)	ST(7)	A(1+5) B(2+6) F(4+5)
2	A(6) B(8) F(9)	A(6)	C(2+4) D(5+3)
3	C(6) B(8) D(8) F(9)	C(6)	D(5+3) J(6+6)
4	B(8) D(8) F(9) J(12)	B(8)	D(6+3) E(5+7)
5	D(8) F(9) E(12) J(12)	D(8)	I(10+1)
6	F(9) I(11) E(12) J(12)	F(9)	A(7+5) G(10+7)
7	I(11) E(12) J(12) G(17)	I(11)	EN(13+0)
8	E(12) J(12) EN(13) G(17)	E(12)	D(11+3) H(8+3)
9	H(11) J(12) EN(13) G(17)	H(11)	I(11+1) EN(10+0)
10	EN(10) J(12) G(17)	EN(10)	

Node	h	h*
ST	7	7
A	5	6
B	6	8
C	4	5
D	3	8
E	**7**	**5**
F	5	9
G	**7**	**6**
H	**3**	**2**
I	1	3
J	**6**	**1**
K	1	2

The **h** function is **not admissible** because for the nodes **E, G, H, and J** the estimated cost for reaching the goal is higher than the actual one.

Exercise 3.19

Writing the solution by using the full path.

Step	Queue	Expanded Path	Children

1	ST(7)	ST(7)	ST-A(1+5)
			ST-B(2+6)
			ST-F(4+5)

2	ST-A(6)
	ST-B(8)
	ST-F(9)
		ST-A(6)	ST-A-C(2+4)
			ST-A-D(5+3)

3	ST-A-C(6)
	ST-A-D(8)
	ST-B(8)
	ST-F(9)
		ST-A-C(6)	ST-A-C-J(6+6)
			ST-A-C-D(5+3)

4	ST-A-D(8)
	ST-B(8)
	ST-F(9)
	ST-A-C-J(12)
		ST-A-D(8)	ST-A-D-I(10+1)

5	ST-B(8)
	ST-F(9)
	ST-A-D-I(11)
	ST-A-C-J(12)
		ST-B(8)	ST-B-D(6+3)
			ST-B-E(5+7)

6	ST-B-D(9)
	ST-F(9)
	ST-A-D-I(11)
	ST-A-C-J(12)
	ST-B-E(12)
		ST-B-D(9)	ST-B-D-I(11+1)

Exercise 3.19

Writing the solution by using the full path – Step 9 – Option 1.

Step	Queue	Expanded Path	Children

7	ST-F(9)
	ST-A-D-I(11)
	ST-A-C-J(12)
	ST-B-E(12)
		ST-F(9)	ST-F-A(7+15)
			ST-F-G(10+7)

8	ST-A-D-I(11)
	ST-A-C-J(12)
	ST-B-E(12)
	ST-F-G(17)
		ST-A-D-I(11)	ST-A-D-I-EN(13+0)

9	ST-A-C-J(12)
	ST-B-E(12)
	ST-A-D-I-EN(13)
	ST-F-G(17)
		ST-A-C-J(12)	ST-A-C-J-EN(7+0)
			ST-A-C-J-I(8+1)

10	**ST-A-C-J-EN(7+0)**
	ST-A-C-J-I(9)
	ST-B-E(12)
	ST-A-D-I-EN(13)
	ST-F-G(17)		

Exercise 3.19

Writing the solution by using the full path – Step 9 – Option 2.

Step	Queue	Expanded Path	Children

7	ST-F(9)
	ST-A-D-I(11)
	ST-A-C-J(12)
	ST-B-E(12)
		ST-F(9)	ST-F-A(7+15)
			ST-F-G(10+7)

8	ST-A-D-I(11)
	ST-A-C-J(12)
	ST-B-E(12)
	ST-F-G(17)
		ST-A-D-I(11)	ST-A-D-I-EN(13+0)

9	ST-B-E(12)
	ST-A-C-J(12)
	ST-A-D-I-EN(13)
	ST-F-G(17)
		ST-B-E(12)	ST-B-E-D(11+3)
			ST-B-E-H(8+3)

10	ST-B-E-H(11)
	ST-A-C-J(12)
	ST-A-D-I-EN(13)
	ST-F-G(17)
		ST-B-E-H(11)	ST-B-E-H-EN(10+0)

11	**ST-B-E-H-EN(10)**
	ST-A-C-J(12)
	ST-F-G(17)		

Exercise 3.19

Step	Queue	Processed Nodes	Children
1	ST(7)	ST(7)	A(1+5) B(2+6) F(4+5)
2	A(6) B(8) F(9)	A(6)	C(2+4) D(5+3)
3	C(6) B(8) D(8) F(9)	C(6)	D(5+3) J(6+6)
4	B(8) D(8) F(9) J(12)	B(8)	D(6+3) E(5+7)
5	D(8) F(9) E(12) J(12)	D(8)	I(10+1)
6	F(9) I(11) E(12) J(12)	F(9)	A(7+5) G(10+7)
7	I(11) E(12) J(12) G(17)	I(11)	EN(13+0)
8	E(12) J(12) EN(13) G(17)	E(12)	D(11+3) H(8+3)
9	H(11) J(12) EN(13) G(17)	H(11)	I(11+1) EN(10+0)
10	EN(10) J(12) G(17)	EN(10)	

Node	h	h*
ST	7	7
A	5	6
B	6	8
C	4	5
D	3	8
E	**7**	**5**
F	5	9
G	**7**	**6**
H	**3**	**2**
I	1	3
J	**6**	**1**
K	1	2

The **h** function is **not admissible** because for the nodes **E, G, H, and J** the estimated cost for reaching the goal is higher than the actual one.

Exercise 3.20

- Maximize the following f(x): x^3 – 60 * x^2 + 900 * x – 100
- Admissible values of x: 0 ≤ x ≤ 31
- Selection: roulette wheel
- Crossover probability: 1.0
- Mutation probability: 0.0
- How to represent x?

Exercise 3.20

- Maximize the following f(x): x^3 – 60 * x^2 + 900 * x – 100
- Admissible values of x: 0 ≤ x ≤ 31
- Selection: roulette wheel
- Crossover probability: 1.0
- Mutation probability: 0.0
- How to represent x? → By using five binary digits.

Chromosome	Binary string	X	f(x)
P1	11100	28	212
P2	01111	15	3475
P3	10111	23	1227
P4	00100	4	2804
		Total	7718
		Average	1929.50

Exercise 3.20

- Maximize the following f(x): x^3 – 60 * x^2 + 900 * x – 100
- Admissible values of x: 0 ≤ x ≤ 31
- Selection: roulette wheel
- Crossover probability: 1.0
- Mutation probability: 0.0
- How to represent x? → By using five binary digits.

Chromosome	Binary string	X	f(x)
P1	11100	28	212
P2	01111	15	**3475**
P3	10111	23	1227
P4	00100	4	2804
		Total	7718
		Average	1929.50

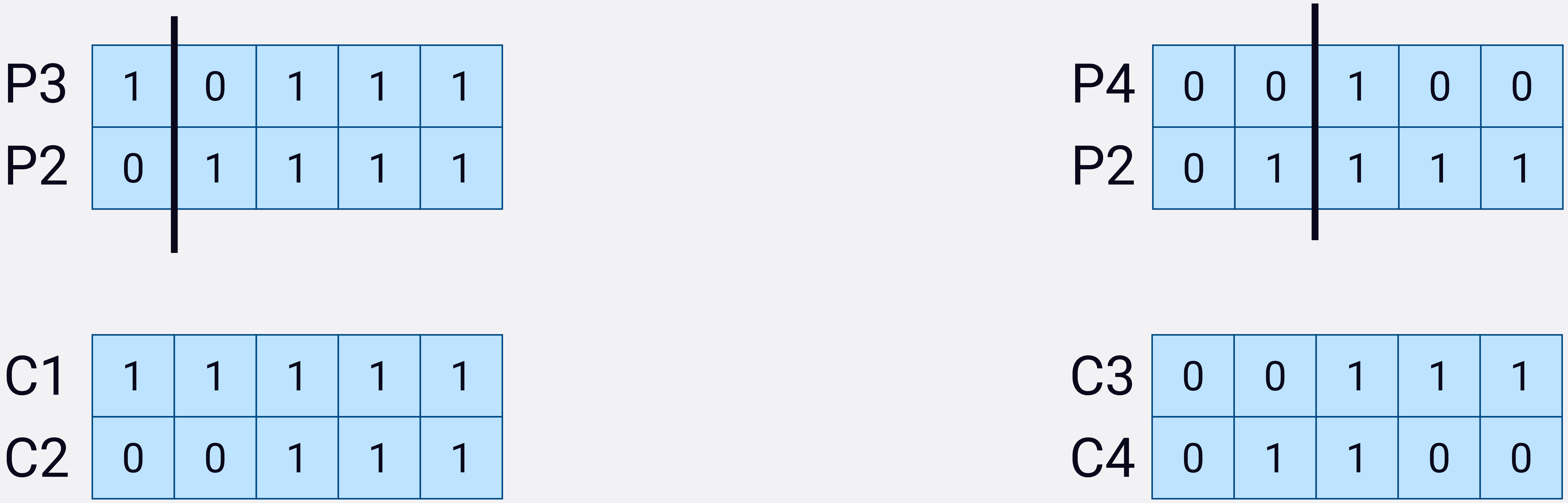
- Selection:
  - 1-212 → P1
  - 213-3687 → P2
  - 3688-4913 → P3
  - 4914-7718 → P4

Round 1: 4416 (P3) and 1925 (P2)
Round 2: 5482 (P4) and 3184 (P2)

Exercise 3.20

- Maximize the following f(x): x^3 – 60 * x^2 + 900 * x – 100
- Admissible values of x: 0 ≤ x ≤ 31
- Selection: roulette wheel
- Crossover probability: 1.0
- Mutation probability: 0.0
- How to represent x? → By using five binary digits.

- Crossover:



P3	1	0	1	1	1
P2	0	1	1	1	1

P4	0	0	1	0	0
P2	0	1	1	1	1

C1	1	1	1	1	1
C2	0	0	1	1	1

C3	0	0	1	1	1
C4	0	1	1	0	0

Exercise 3.20

- Maximize the following f(x): x^3 – 60 * x^2 + 900 * x – 100
- Admissible values of x: 0 ≤ x ≤ 31
- Selection: roulette wheel
- Crossover probability: 1.0
- Mutation probability: 0.0
- How to represent x? → By using five binary digits.

Chromosome	Binary string	X	f(x)
P1	11111	31	131
P2	00111	7	3803
P3	00111	7	3803
P4	01100	12	**3889**
		Total	11735
		Average	2933.75

Exercise 3.21

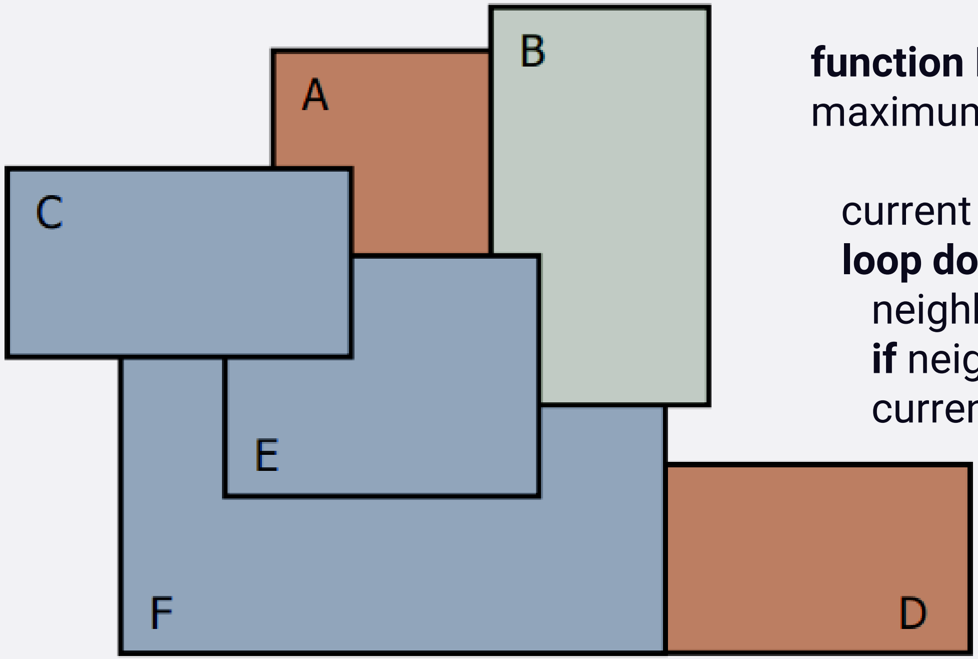
- Maximize the following f(x): x^3 – 60 * x^2 + 900 * x – 100
- Admissible values of x: 0 ≤ x ≤ 63
- Selection: roulette wheel
- Crossover probability: 0.9
- Mutation probability: 0.05

Chromosome	Binary string	X	f(x)
P1	101001	41	4861
P2	011011	27	143
P3	101011	43	7167
P4	010100	20	1900
		Total	14071
		Average	3517.75

Exercise 4.3

- Graph coloring: start with random coloring of nodes and change color of one node to reduce the number of conflicts.

Step	A	B	C	D	E	F	# conflicts
1	r	g	b	r	b	b	3 {CE, CF, EF}

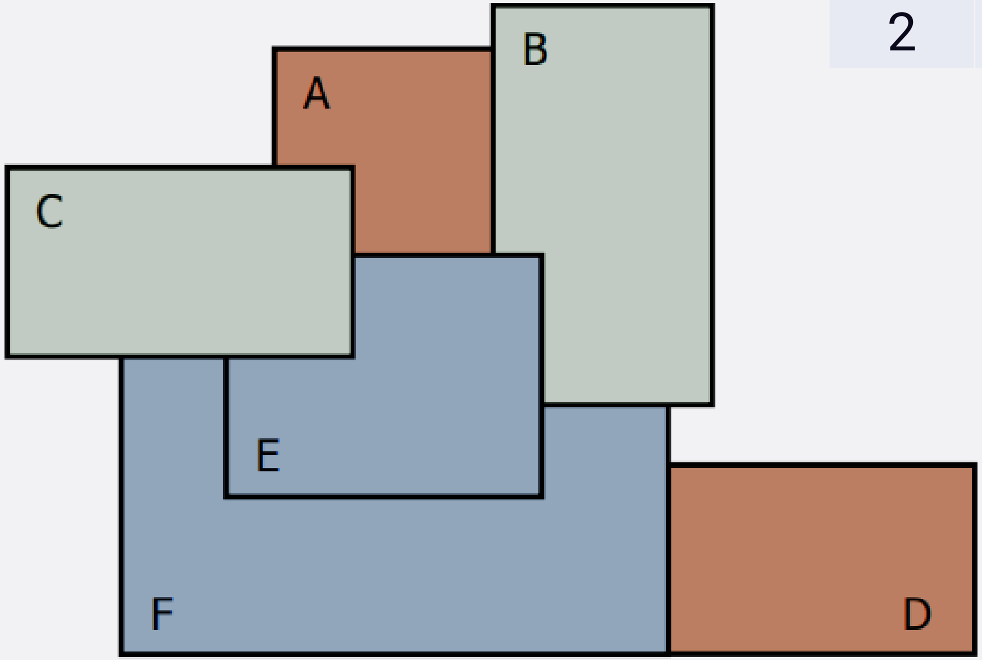


```
function HILL-CLIMBING (problem) returns a state that is a local maximum

  current ← MAKE-NODE (problem.INITIAL-STATE)
  loop do
    neighbor ← a highest-valued successor of current
    if neighbor.VALUE ≤ current.VALUE then return current.STATE
    current ← neighbor
```

Exercise 4.3

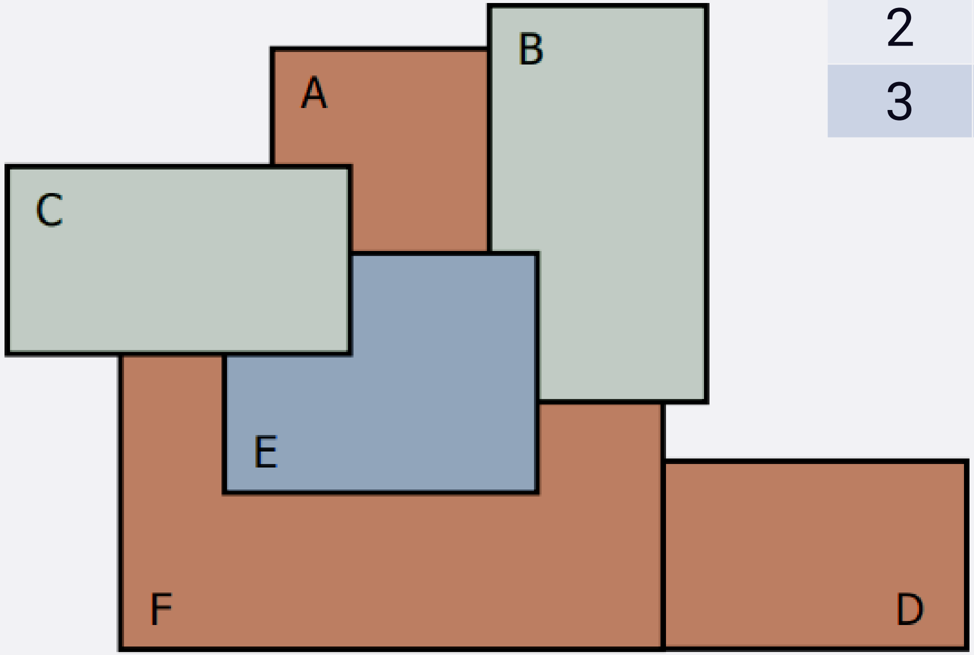
- Graph coloring: start with random coloring of nodes and change color of one node to reduce the number of conflicts.



Step	A	B	C	D	E	F	# conflicts
1	r	g	b	r	b	b	3 {CE, CF, EF}
2	r	g	**G**	r	b	b	1 {EF}

Exercise 4.3

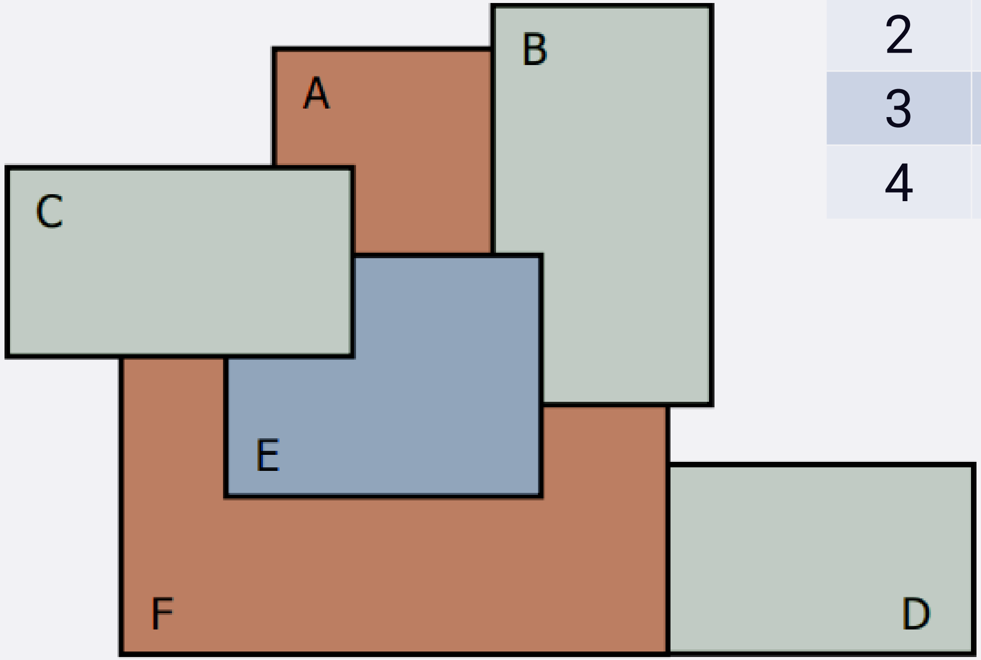
- Graph coloring: start with random coloring of nodes and change color of one node to reduce the number of conflicts.



Step	A	B	C	D	E	F	# conflicts
1	r	g	b	r	b	b	3 {CE, CF, EF}
2	r	g	**G**	r	b	b	1 {EF}
3	r	g	g	r	b	**R**	1 {DF}

Exercise 4.3

- Graph coloring: start with random coloring of nodes and change color of one node to reduce the number of conflicts.



Step	A	B	C	D	E	F	# conflicts
1	r	g	b	r	b	b	3 {CE, CF, EF}
2	r	g	**G**	r	b	b	1 {EF}
3	r	g	g	r	b	**R**	1 {DF}
4	r	g	g	**G**	b	r	0