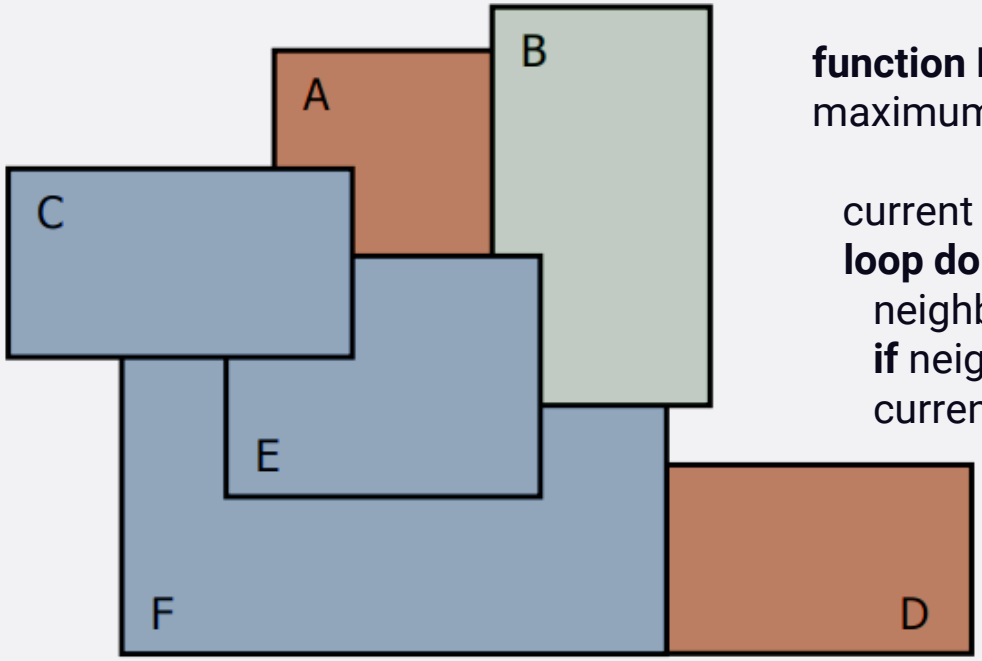
Exercise 4.3

- 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}



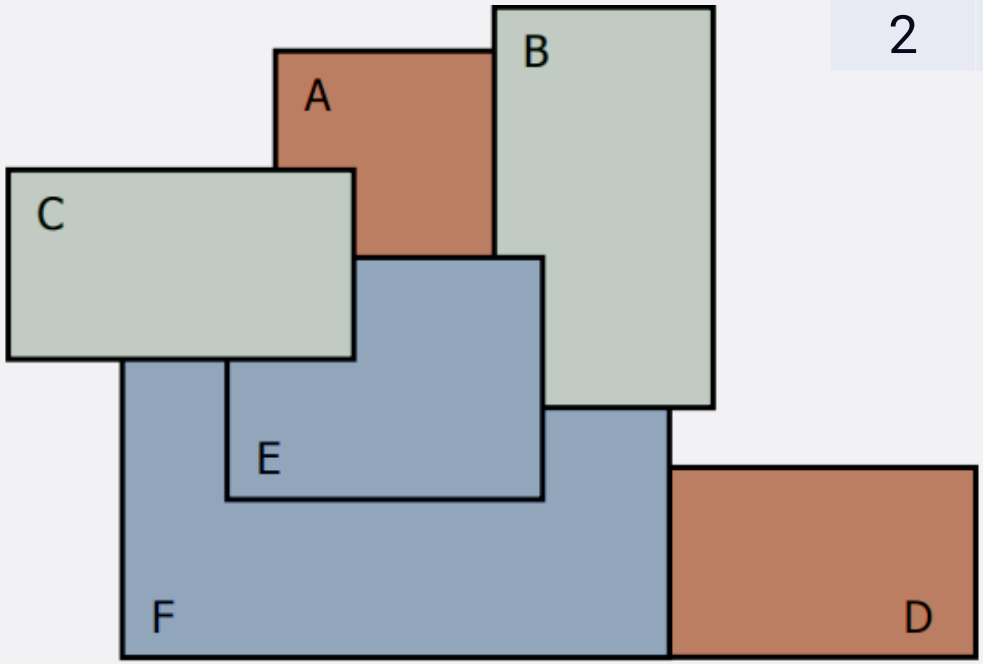
function HILL-CLIMBING (problem) **returns** a state that is a local maximum

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

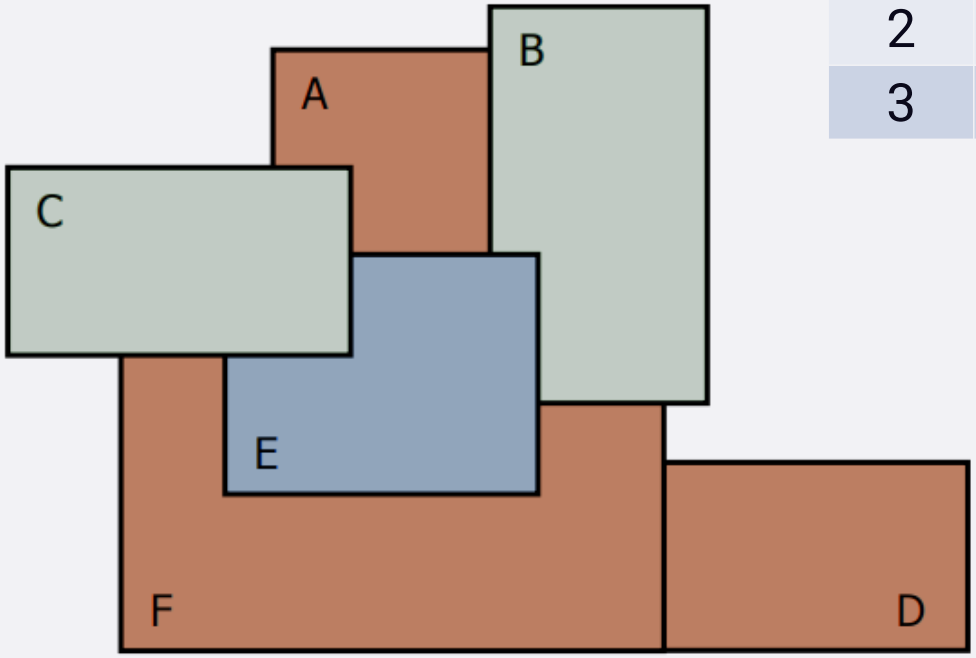
- Graph coloring: start with random coloring of nodes and change color of one node to reduce the number of conflicts.

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

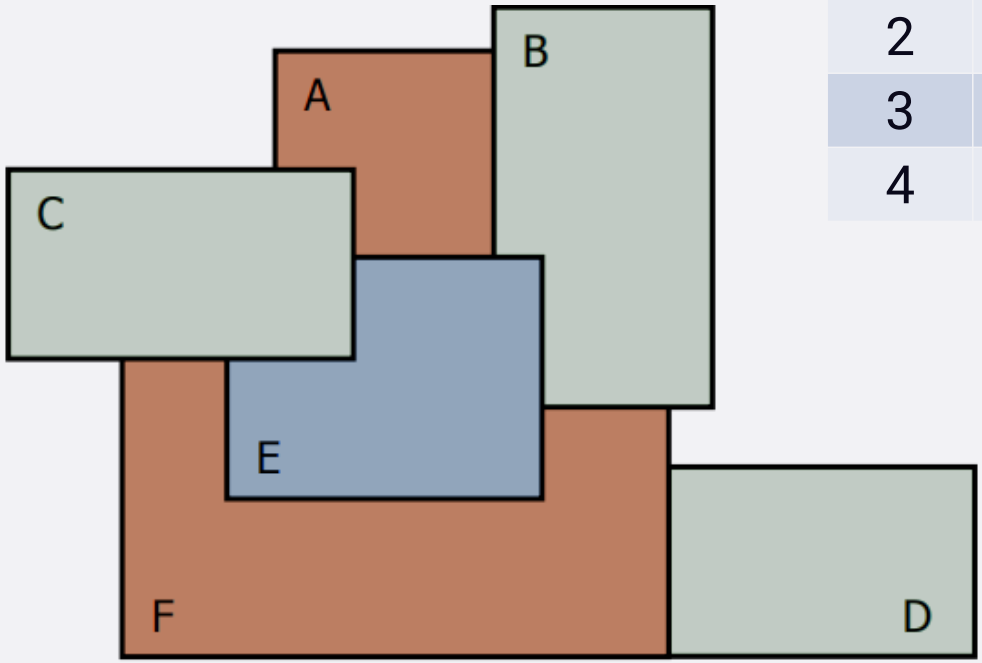
- Graph coloring: start with random coloring of nodes and change color of one node to reduce the number of conflicts.



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