Graph Coloring

Exact and Approximate Algorithms

1705039 , 1705044

Group-7

Problem Statement

Solution Overview

Greedy Algorithm

Dynamic Programming

Applications

Assign colors to *certain elements* of a graph subject to *certain constraints*

Basic Definitions

Assign colors to *certain elements* of a graph subject to *certain constraints*

Vertex coloring is the most common graph coloring problem.

Assign colors to *certain elements* of a graph subject to *certain constraints*

Vertex coloring is the most common graph coloring problem.

"A way of coloring the vertices of a graph such that no two adjacent vertices are of the same color."

Assign new colors
for every vertex

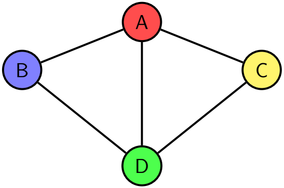


Figure: 4 colors

Find the minimum colors
(*chromatic number*)

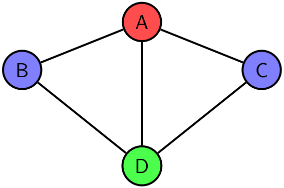


Figure: 3 colors

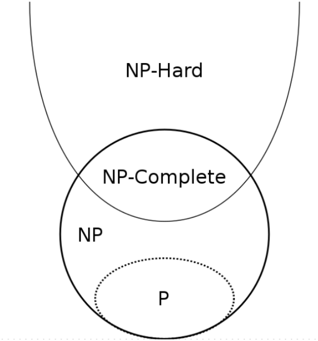
Problem Statement

Solution Overview

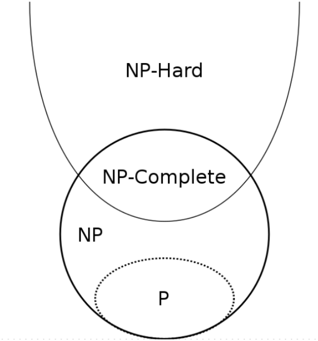
Greedy Algorithm

Dynamic Programming

Applications



Decision problem
(is the graph k-colorable?)
is **NP-Complete**.



Decision problem
(is the graph k-colorable?)
is **NP-Complete**.

Optimization problem
(find minimum colors k)
is **NP-Hard**.

Approximate algorithm

- Greedy method
- Solvable in limited time
- May not yield minimum

Approximate algorithm

- Greedy method
- Solvable in limited time
- May not yield minimum

Exact algorithm

- Dynamic Programming
- Minimum guaranteed
- Under constraints

Basic Idea

Make *locally optimal choice* at each step

Greedy choice: *Using existing colors*

Make *locally optimal choice* at each step

Greedy choice: *Using existing colors*

- Reuse a color k

$$V_i.color = k : (k \in C) \ \& \ (k \notin \varepsilon) \tag{1}$$

Make *locally optimal choice* at each step

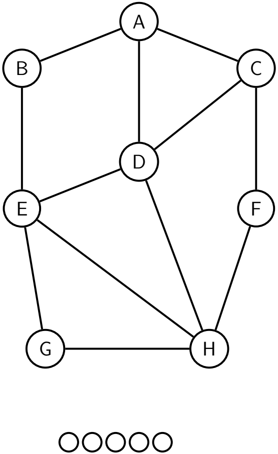
Greedy choice: *Using existing colors*

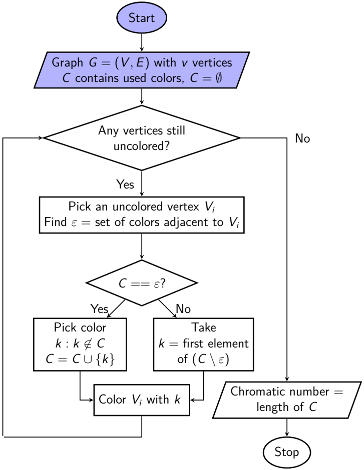
- Reuse a color k

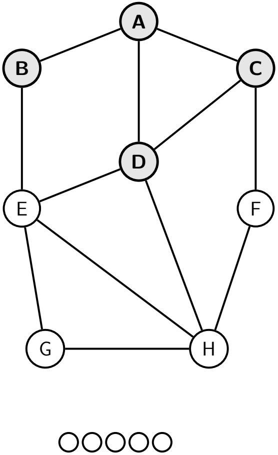
$$V_i.color = k : (k \in C) \ \& \ (k \notin \varepsilon) \tag{1}$$

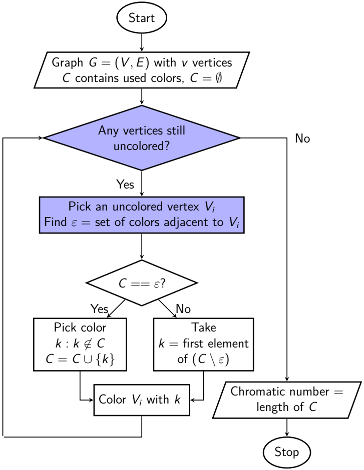
- If colors are exhausted

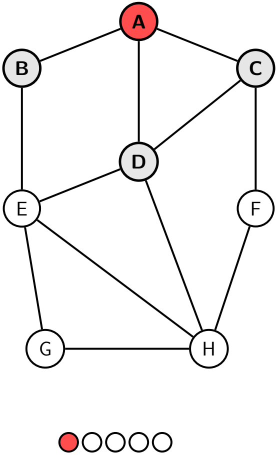
$$\begin{gathered} V_i.color = k : (k \notin C) \\ C = C \cup \{k\} \end{gathered} \tag{2}$$

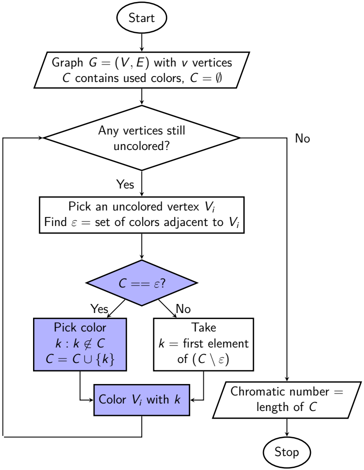

A
B
C
D
E
F
G
H

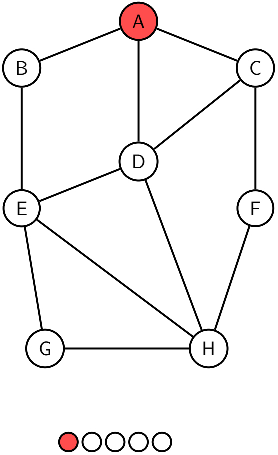

Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find $\varepsilon =$ set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color
$k : k \notin C$
$C = C \cup \{k\}$
Take
$k =$ first element
of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number $=$
length of C
Stop

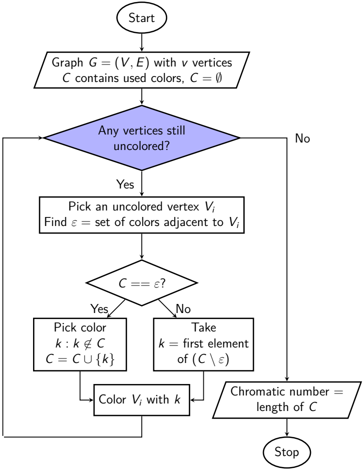

A
B
C
D
E
F
G
H

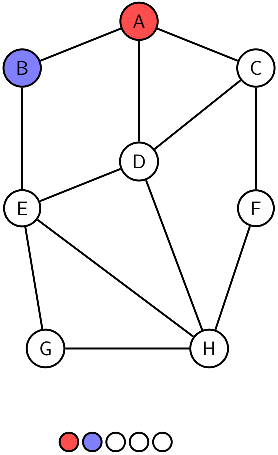

Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find ε = set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color $k : k \notin C$ $C = C \cup \{k\}$
Take k = first element of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number = length of C
Stop

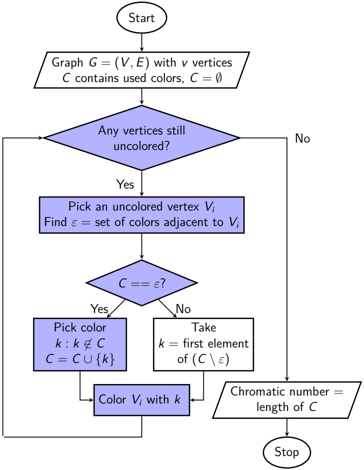

A
B
C
D
E
F
G
H

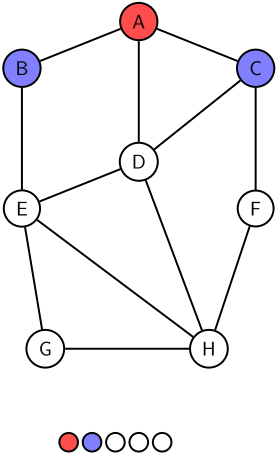

Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find ε = set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color
$k : k \notin C$
$C = C \cup \{k\}$
Take
k = first element
of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number = length of C
Stop

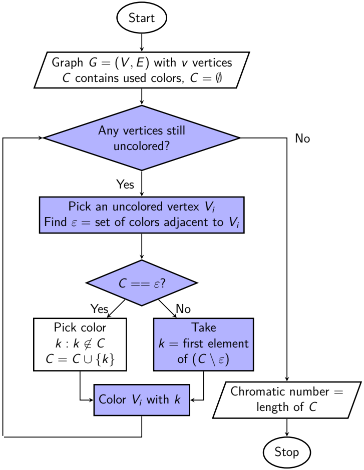

A
B
C
D
E
F
G
H

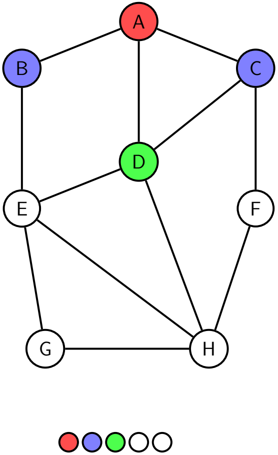

Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find $\varepsilon =$ set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color $k : k \notin C$ $C = C \cup \{k\}$
Take $k =$ first element of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number = length of C
Stop

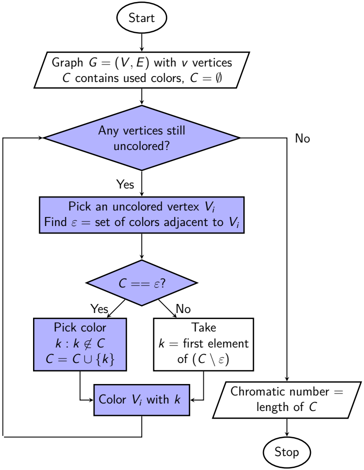

A
B
C
D
E
F
G
H


Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find $\varepsilon =$ set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color
$k : k \notin C$
$C = C \cup \{k\}$
Take
$k =$ first element of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number = length of C
Stop

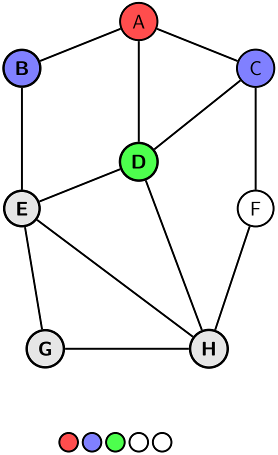

A
B
C
D
E
F
G
H

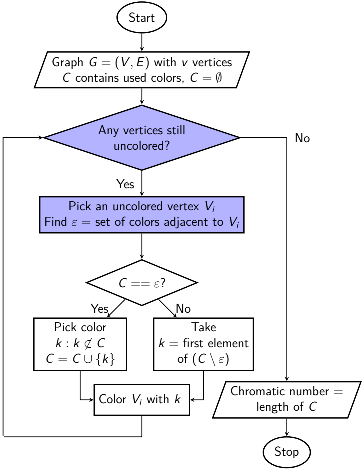

Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find $\varepsilon =$ set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color
$k : k \notin C$
$C = C \cup \{k\}$
Take
$k =$ first element
of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number = length of C
Stop

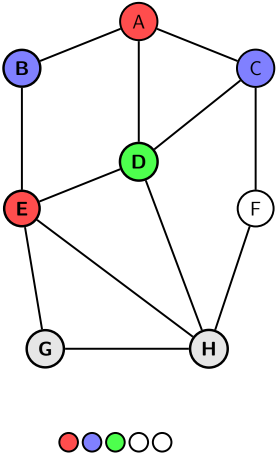

A
B
C
D
E
F
G
H

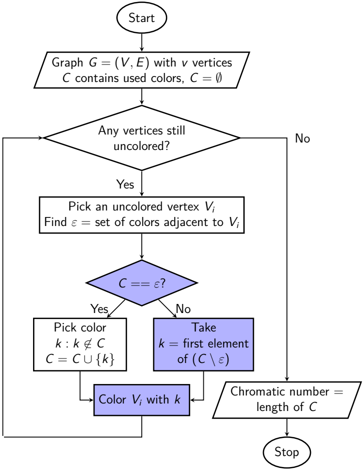

Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find $\varepsilon =$ set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color
$k : k \notin C$
$C = C \cup \{k\}$
Take
$k =$ first element of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number = length of C
Stop

Simulation

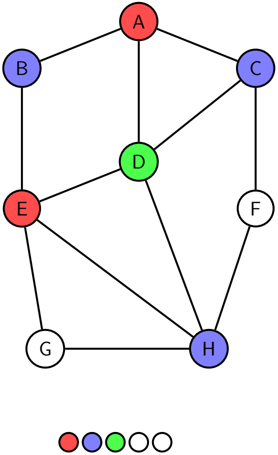

A
B
C
D
E
F
G
H

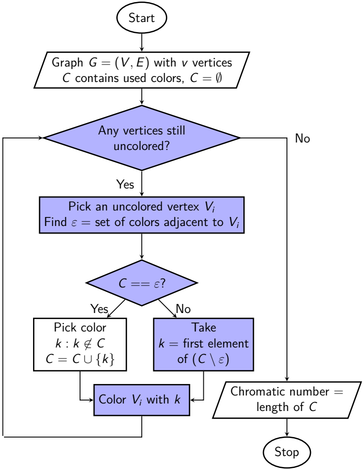

Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find ε = set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color $k : k \notin C$ $C = C \cup \{k\}$
Take k = first element of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number = length of C
Stop

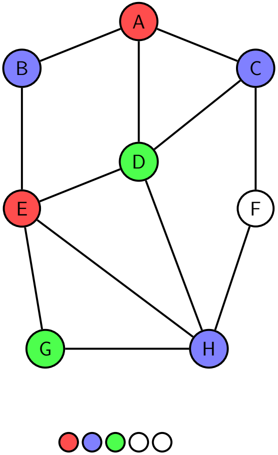

A
B
C
D
E
F
G
H

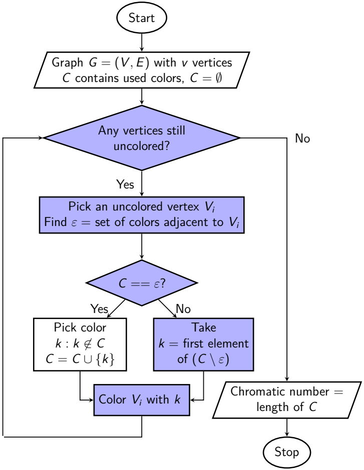

Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find $\varepsilon =$ set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color
$k : k \notin C$
$C = C \cup \{k\}$
Take
$k =$ first element
of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number $=$
length of C
Stop

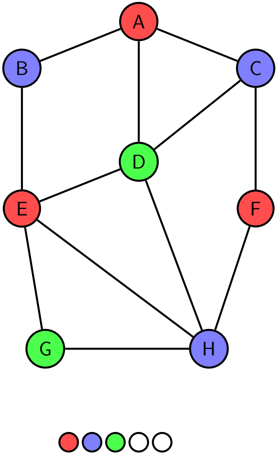
Simulation

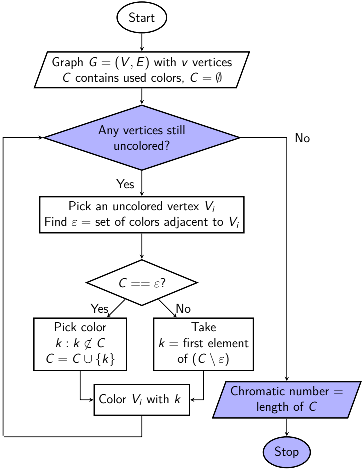

A
B
C
D
E
F
G
H


Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find $\varepsilon =$ set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color
$k : k \notin C$
$C = C \cup \{k\}$
Take
$k =$ first element of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number = length of C
Stop


A
B
C
D
E
F
G
H


Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find $\varepsilon =$ set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color
$k : k \notin C$
$C = C \cup \{k\}$
Take
$k =$ first element
of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number $=$
length of C
Stop


A
B
C
D
E
F
G
H


Start
Graph $G = (V, E)$ with v vertices
C contains used colors, $C = \emptyset$
Any vertices still uncolored?
No
Yes
Pick an uncolored vertex V_i
Find ε = set of colors adjacent to V_i
$C == \varepsilon$?
Yes
No
Pick color
$k : k \notin C$
$C = C \cup \{k\}$
Take
k = first element
of $(C \setminus \varepsilon)$
Color V_i with k
Chromatic number = length of C
Stop

- Time Complexity : $O(V^2 + E)$

Complexity & Limitations

- Time Complexity : $O(V^2 + E)$
- Doesn't guarantee minimum number of colors

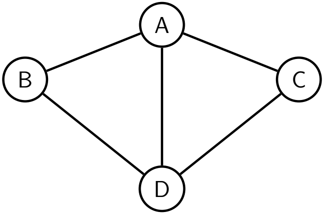
Complexity & Limitations

- Time Complexity : $O(V^2 + E)$
- Doesn't guarantee minimum number of colors
- Upper bound of $d + 1$ where d is maximum degree

Combine *solutions of sub-problems*

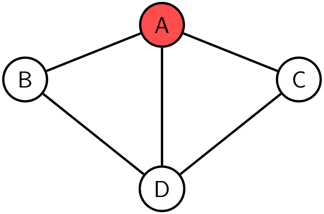
Combine *solutions of sub-problems*

Apply: Chromatic number of a graph is derived from its sub-graphs



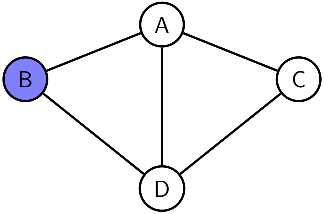
Combine *solutions of sub-problems*

Apply: Chromatic number of a graph is derived from its sub-graphs



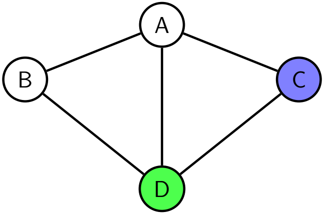
Combine *solutions of sub-problems*

Apply: Chromatic number of a graph is derived from its sub-graphs



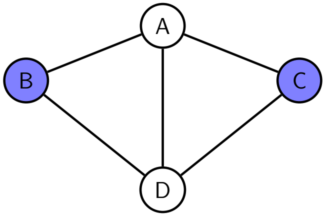
Combine *solutions of sub-problems*

Apply: Chromatic number of a graph is derived from its sub-graphs



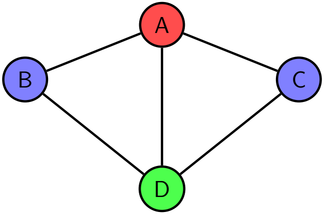
Combine *solutions of sub-problems*

Apply: Chromatic number of a graph is derived from its sub-graphs



Combine *solutions of sub-problems*

Apply: Chromatic number of a graph is derived from its sub-graphs



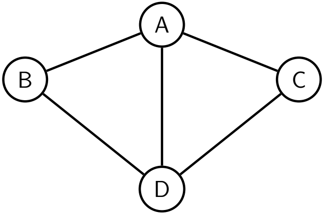
Basic Idea

Idea: A *maximal independent set* is 1-colorable

Idea: A *maximal independent set* is 1-colorable

Solution:

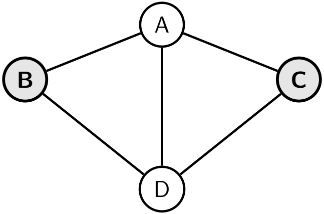
$$\chi(G[S]) = 1 + \chi(G[S \setminus I])$$



Idea: A *maximal independent set* is 1-colorable

Solution:

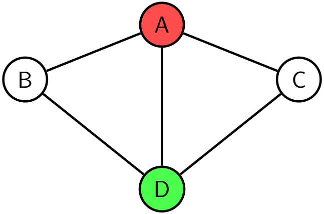
$$\chi(G[S]) = 1 + \chi(G[S \setminus I])$$



Idea: A *maximal independent set* is 1-colorable

Solution:

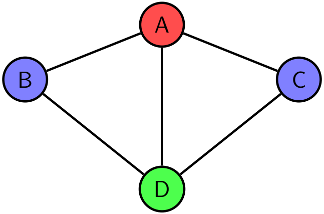
$$\chi(G[S]) = 1 + \chi(G[S \setminus I])$$



Idea: A *maximal independent set* is 1-colorable

Solution:

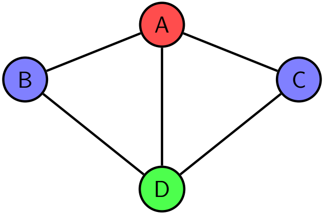
$$\chi(G[S]) = 1 + \chi(G[S \setminus I])$$



Idea: A *maximal independent set* is 1-colorable

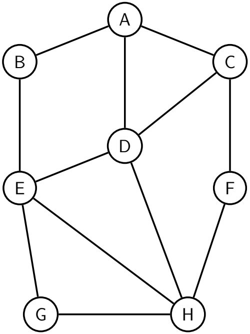
Solution:

$$\chi(G[S]) = 1 + \chi(G[S \setminus I])$$



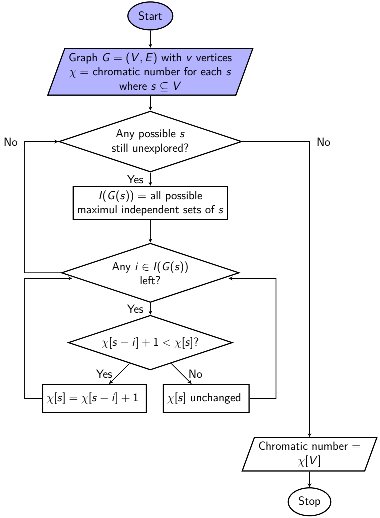
But there is a catch!

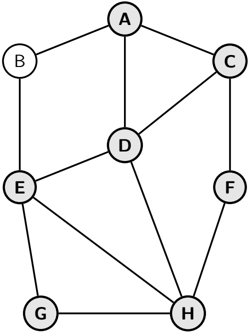
Simulation



χ								
	3	2	2	2	X	X	X	X

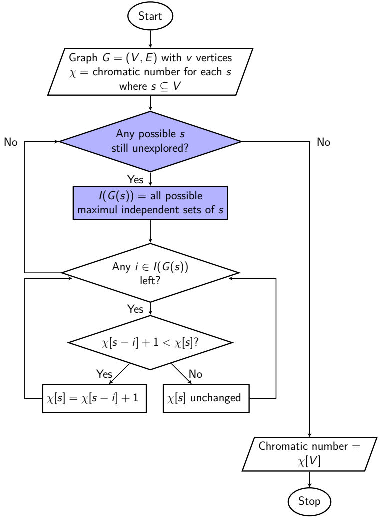
$s - i$			
	E,D,C,H	A,C,E,H	G,D,C,H

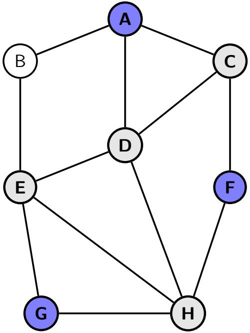




χ	3	2	2	2	X	X	X	X

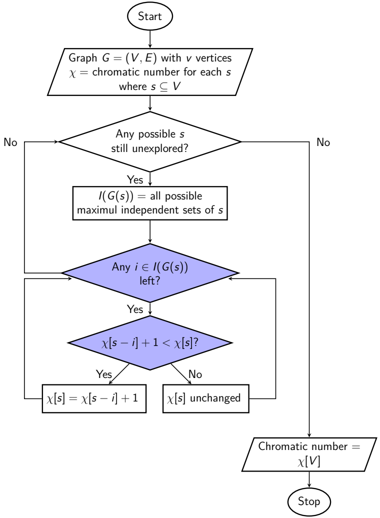
$s - i$	E,D,C,H	A,C,E,H	G,D,C,H

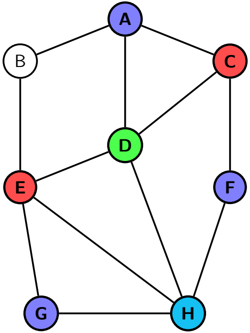




χ	3	2	2	2	X	X	X	X

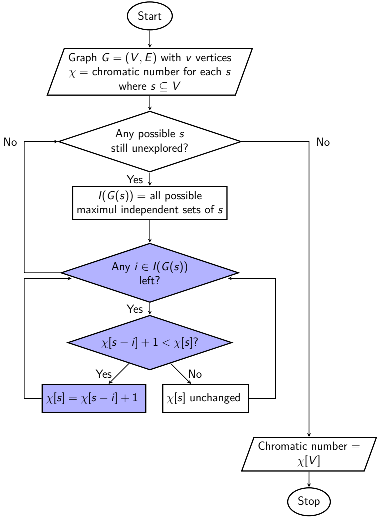
$s-i$	E,D,C,H	A,C,E,H	G,D,C,H

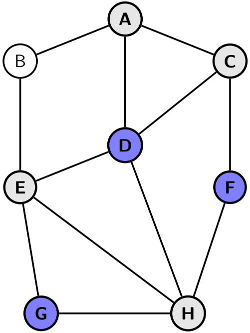




χ								
3	2	2	2	4	X	X	X	

$s - i$		
E,D,C,H	A,C,E,H	G,D,C,H



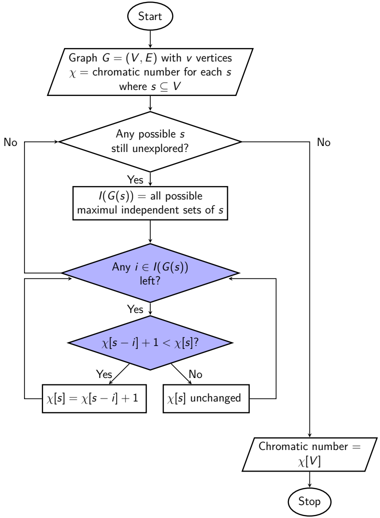


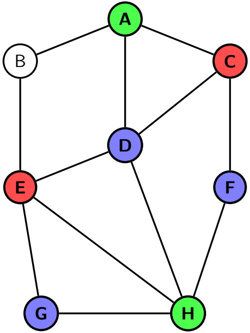
χ

3	2	2	2	4	X	X	X

$s - i$

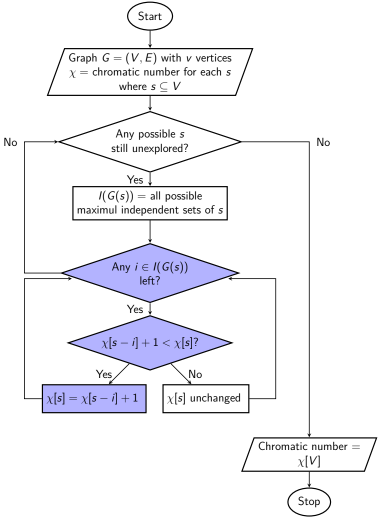
E,D,C,H	A,C,E,H	G,D,C,H

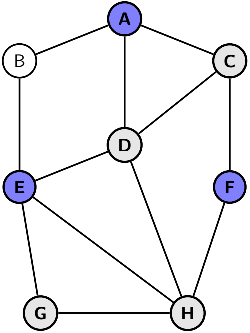




χ	3	2	2	2	3	X	X	X

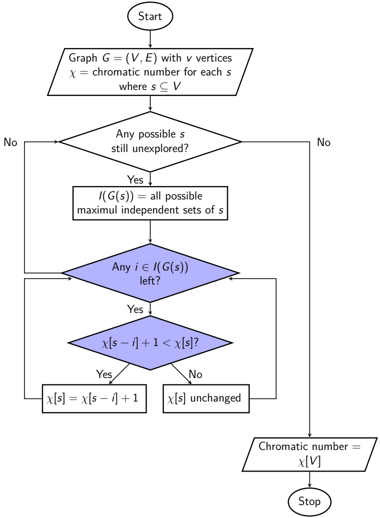
$s - i$	E,D,C,H	A,C,E,H	G,D,C,H

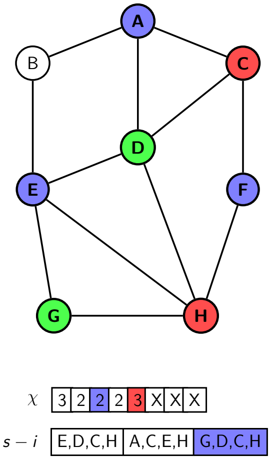


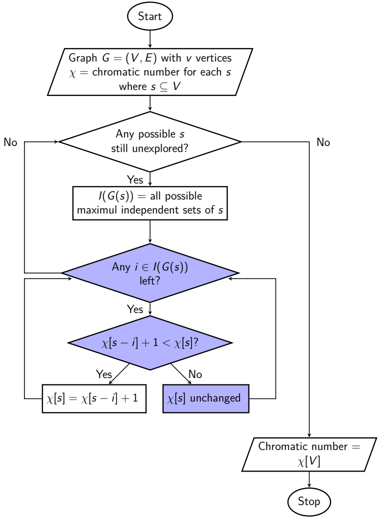


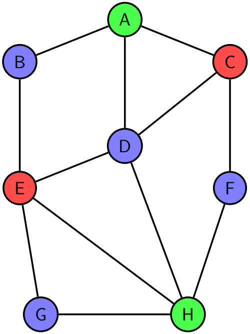
χ								
3	2	2	2	3	X	X	X	

$s - i$		
E,D,C,H	A,C,E,H	G,D,C,H



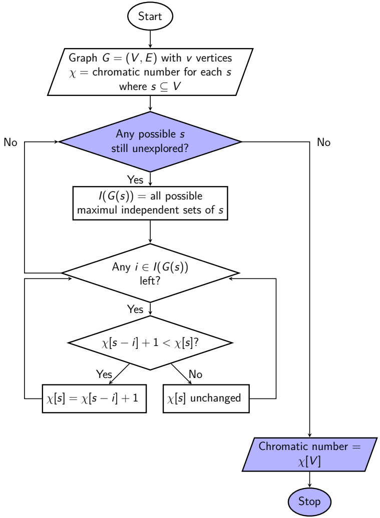

A
B
C
D
E
F
G
H
χ
3
2
2
2
3
X
X
X
s − i
E,D,C,H
A,C,E,H
G,D,C,H


Start
Graph G = (V, E) with v vertices
χ = chromatic number for each s
where s ⊆ V
Any possible s
still unexplored?
No
No
Yes
I(G(s)) = all possible
maximul independent sets of s
Any i ∈ I(G(s))
left?
Yes
χ[s − i] + 1 < χ[s]?
Yes
No
χ[s] = χ[s − i] + 1
χ[s] unchanged
Chromatic number =
χ[V]
Stop



χ	3	2	2	2	3	3	3	3

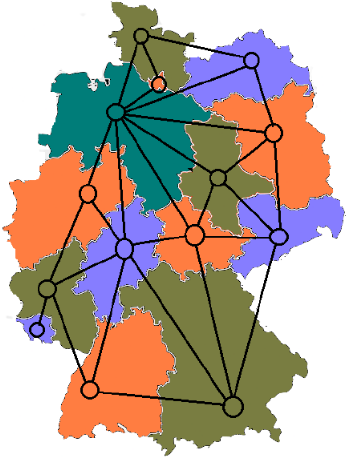
$s - i$	E,D,C,H	A,C,E,H	G,D,C,H



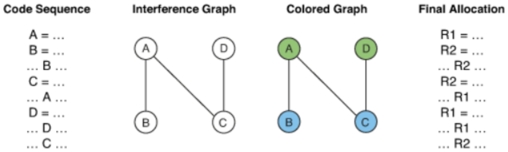
- Time Complexity: $O(2.4423^n)$

- Time Complexity: $O(2.4423^n)$
- Modifications by **Epstein** and **Byskov** lowers complexity to $O(2.4023^n)$

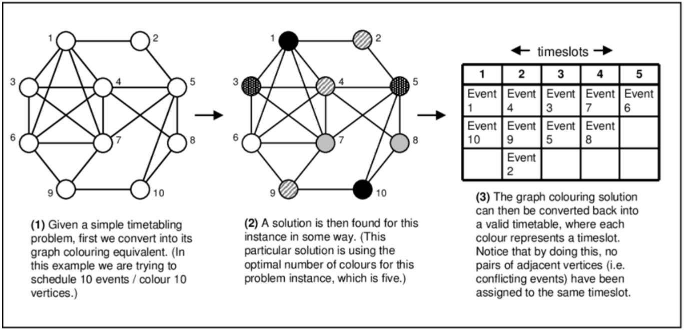
- Time Complexity: $O(2.4423^n)$
- Modifications by **Epstein** and **Byskov** lowers complexity to $O(2.4023^n)$
- All algorithms require exponential space



Coloring geographical maps of countries or states



Assigning variables onto CPU registers



1	2	3	4	5
Event 1	Event 4	Event 3	Event 7	Event 6
Event 10	Event 9	Event 5	Event 8	
	Event 2			

(1) Given a simple timetabling problem, first we convert into its graph colouring equivalent. (In this example we are trying to schedule 10 events / colour 10 vertices.)

(2) A solution is then found for this instance in some way. (This particular solution is using the optimal number of colours for this problem instance, which is five.)

(3) The graph colouring solution can then be converted back into a valid timetable, where each colour represents a timeslot. Notice that by doing this, no pairs of adjacent vertices (i.e. conflicting events) have been assigned to the same timeslot.

Assigning timeslots with constraints [1]

Bibliography

- Alane Marie de Lima and Renato Carmo. "Exact Algorithms for the Graph Coloring Problem". In: *Revista de Informática Teórica e Aplicada* 25 (Nov. 2018), p. 57.
- E.L. Lawler. "A note on the complexity of the chromatic number problem". In: *Information Processing Letters* 5.3 (1976), pp. 66–67. ISSN: 0020-0190.
- David Eppstein. "Small Maximal Independent Sets and Faster Exact Graph Coloring". In: *CoRR* cs.DS/0011009 (2000).

The End

Thank you!

Any Questions?