Fast Solutions to CSPs

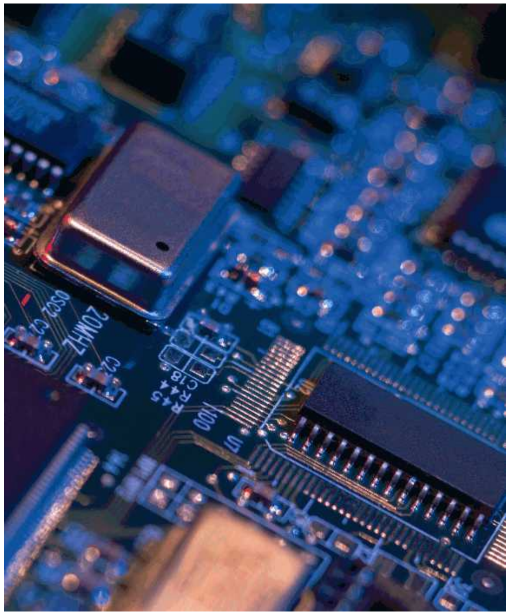
Presented by: Robert Effinger
Dan Lovell

Presented To: 16.412J Cognitive Robotics
MIT

References: "Dynamic Backtracking"
Matthew L. Ginsberg, CIRL, University of Oregon
Journal of Artificial Intelligence Research 1 (1993)
p. 25-46

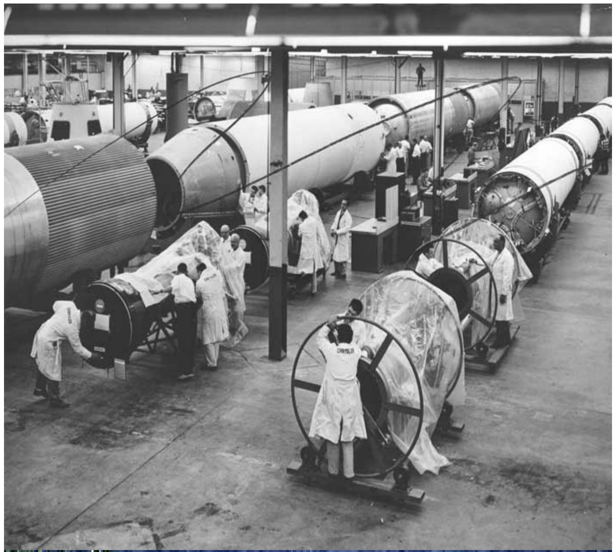
"Hybrid algorithms for the constraint satisfaction problem"
Prosser, P. Computational Intelligence 9 (1993), 268-299.

April 5, 2004



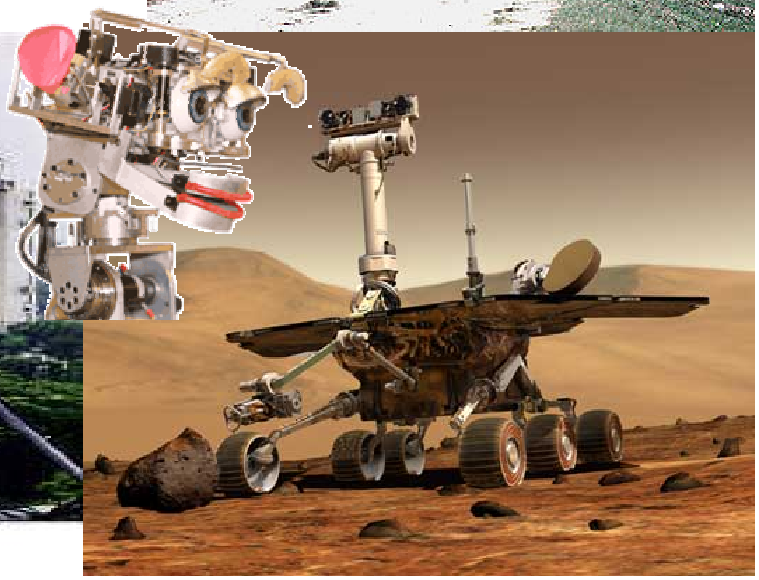
Motivation

- Mobile robot planning
- Resource scheduling
- laying out a silicon chip
- interpret a visual image
- manufacturing processes
- design of airline timetable
- radio frequency planning



Arecibo Observatory in Puerto Rico

A microwave Tower in Malaysia



Quick Definition of a CSP

Constraint Satisfaction Problem (I,V,C)

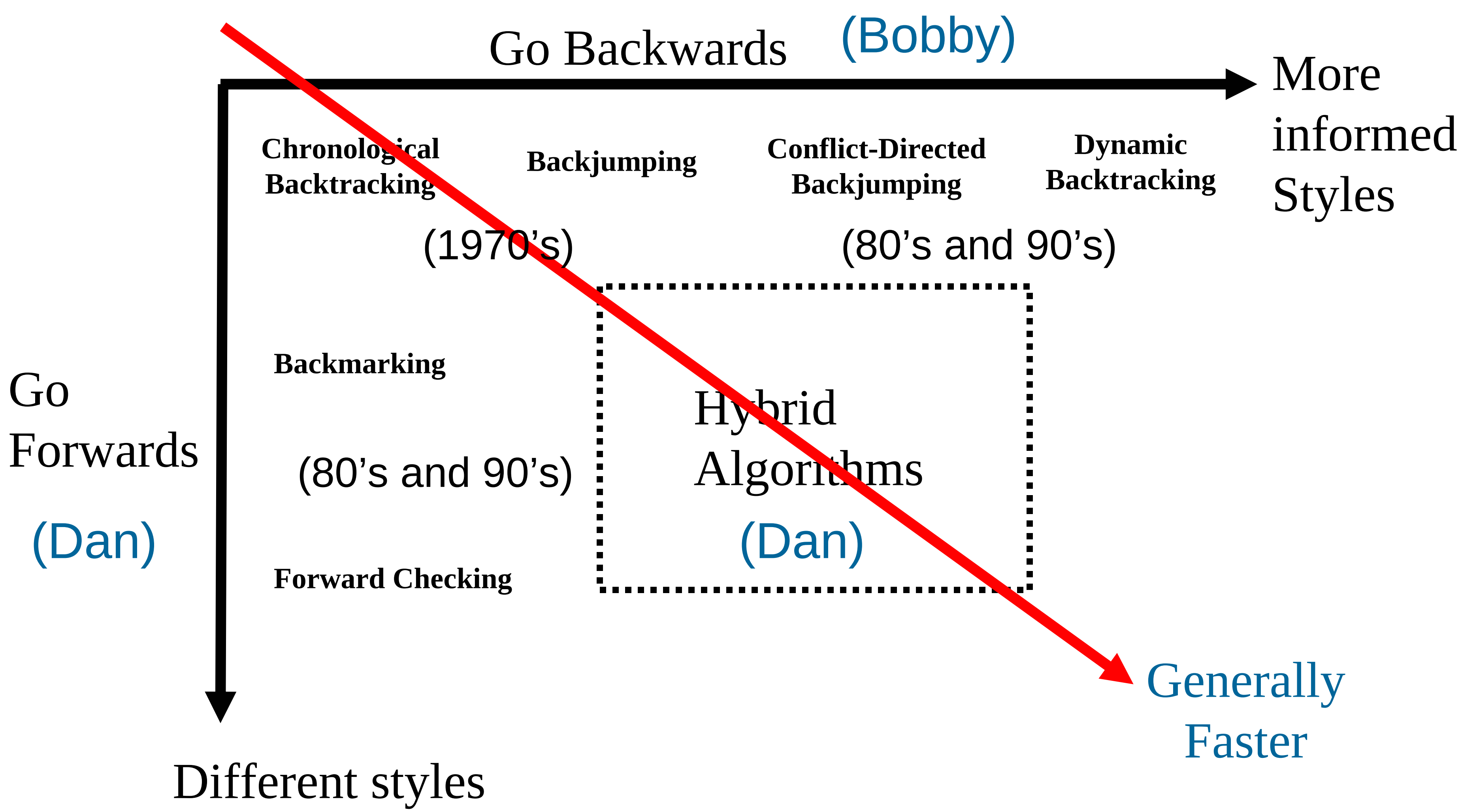
- I , a set of variables
- Vi, a set of possible values for each variable in I.
- C, a set of Cij constraints, each a binary relation

$C = \{C1,1 \ldots C1,n \;\; C2,1 \ldots C2,n \ldots Cn,n\}$

A Solution is found when each variable I is assigned a value from it's domain Vi and the set of all Constraints {C} is satisfied.

How our two talks fit together

Six base styles of CSP search



Dynamic Backtracking

and a review of: Chronological Backtracking and
Conflict-Directed Backjumping

Advanced Lecture Topic: Fast Solutions to CSPs

Presented by: Robert Effinger

Presented To: 16.412J Cognitive Robotics
MIT

Reference: "Dynamic Backtracking"
Matthew L. Ginsberg, CIRL, University of Oregon
Journal of Artificial Intelligence Research 1 (1993)
p. 25-46

April 5, 2004

Overview

- Definition of a CSP
- Simple Map Coloring Example
  - Representing a CSP as a Search Tree
  - Introduce the Example Problem
- Compare Three Backtracking Algorithms
  - Chronological Backtracking
  - Conflict-Directed Backjumping
  - Dynamic Backtracking
- Summary of Dynamic Backtracking
  - Pros and Cons

Quick Definition of a CSP

Constraint Satisfaction Problem (I,V,C)

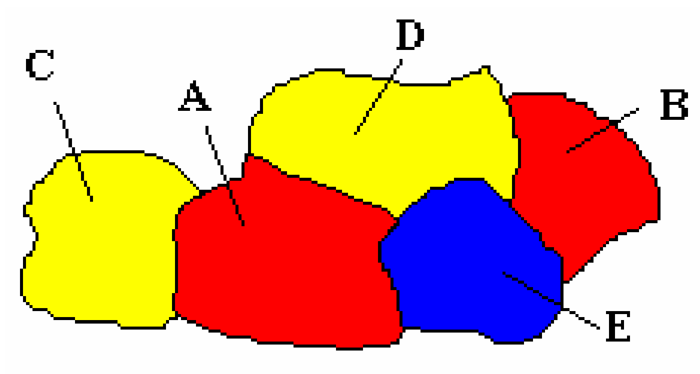
- I , a set of variables
- Vi, a set of possible values for each variable in I.
- C, a set of Cij constraints, each a binary relation
  $C = \{C1,1 \ldots C1,n \;\; C2,1 \ldots C2,n \ldots Cn,n\}$

A Solution is found when each variable I is assigned a value from it's domain Vi and the set of all Constraints {C} is satisfied.

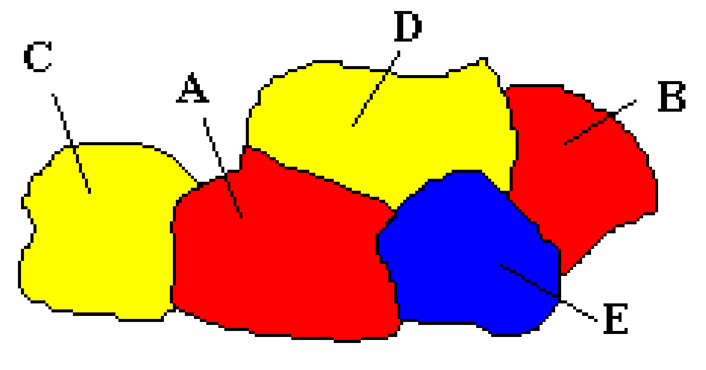
I = {A,B,C,D,E}

Vi = {red,yellow,blue}

Cij = (no neighbor can be the same color)

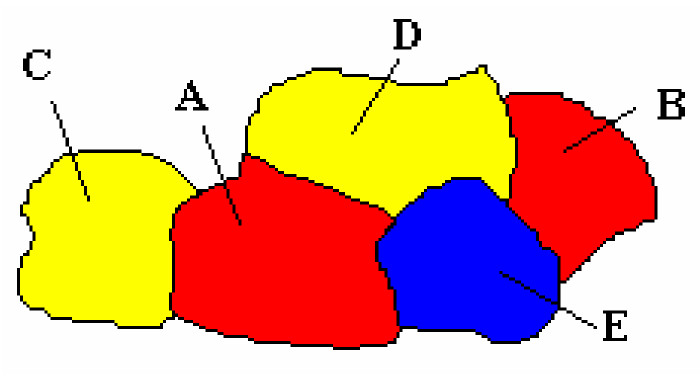


Simple Example Problem



Search Tree Representation of a CSP

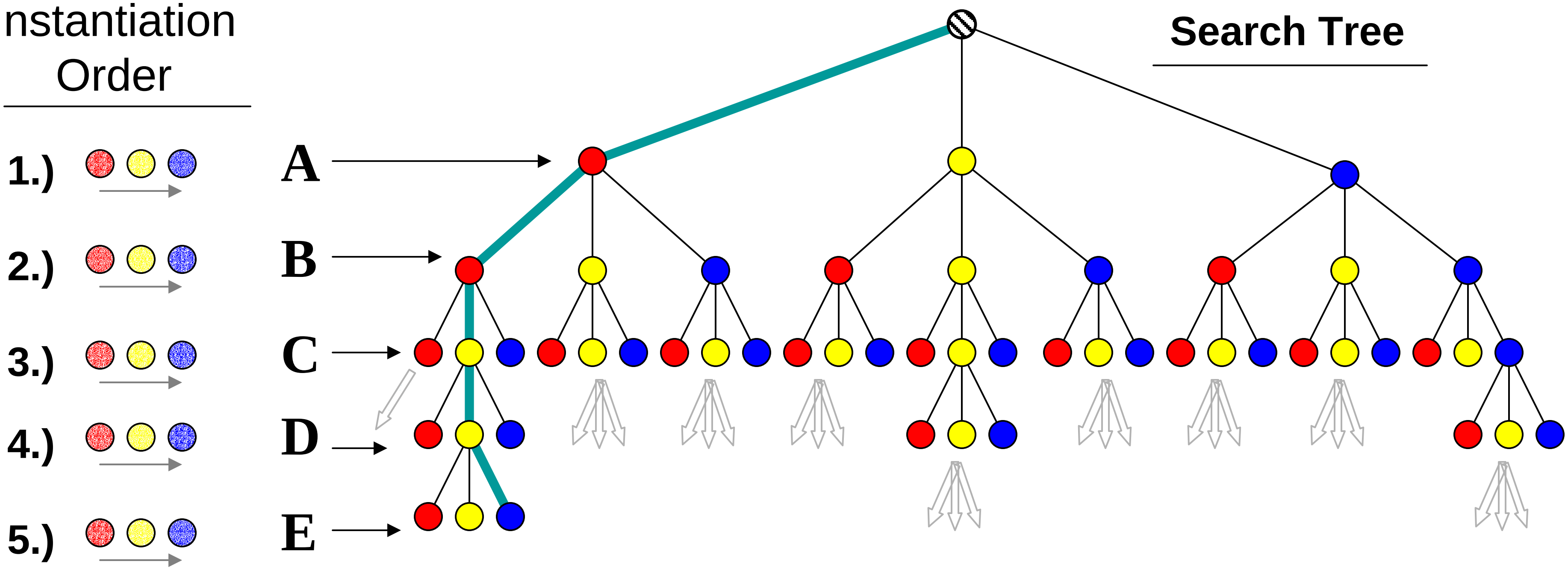
Simple Map Coloring Example



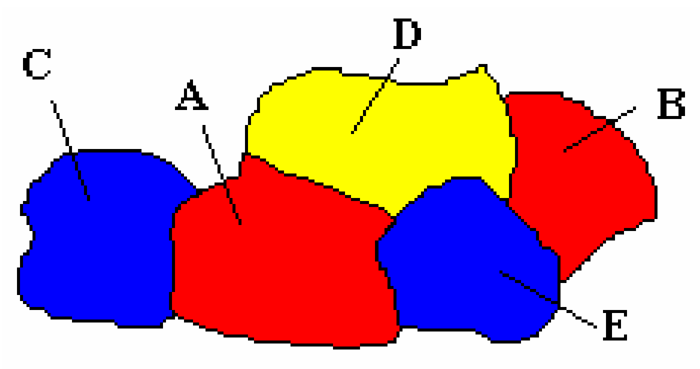
- Variables are assigned values according to an instantiation order
- The search tree grows downward as until each variable is assigned a value from it's domain.
- Dynamic Backtracking allows a dynamic instantiation order

Instantiation Order

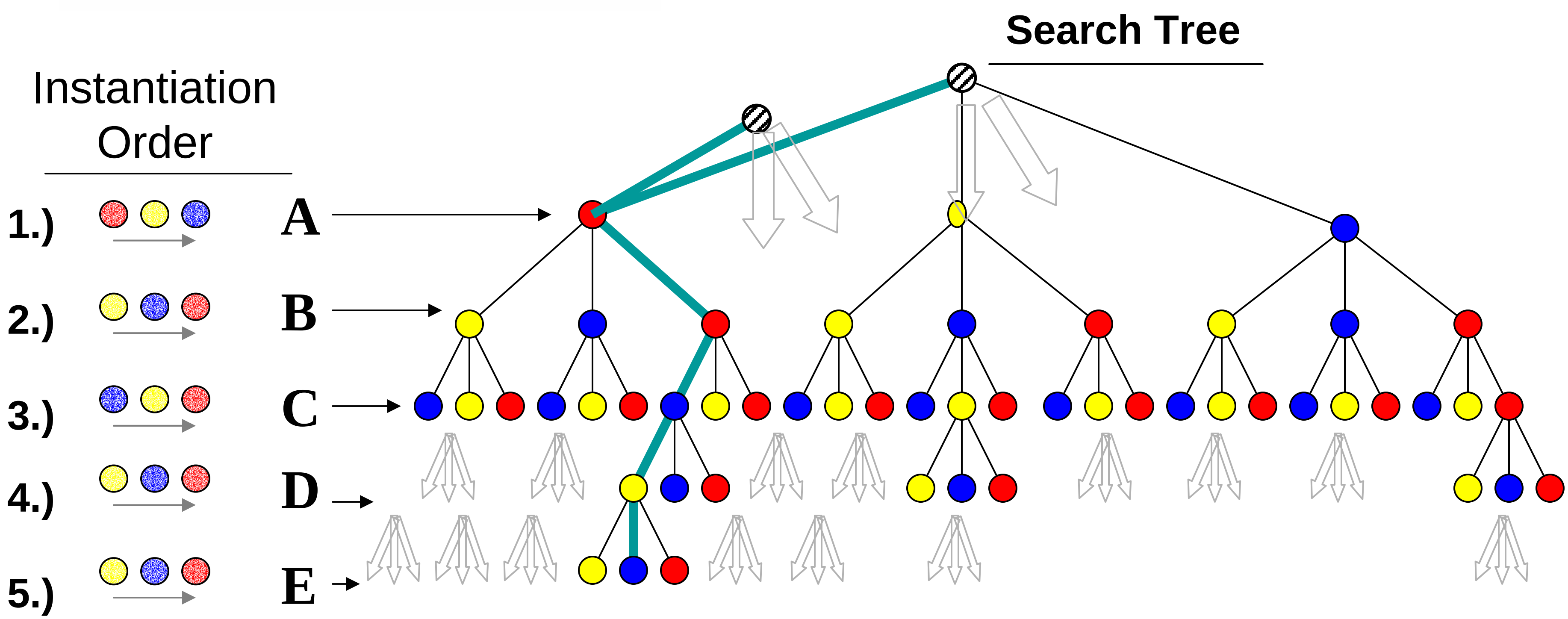
Search Tree



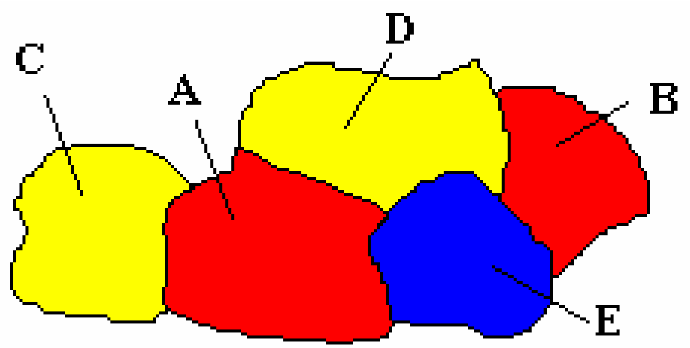
Changing the Color Ordering to Create an Interesting Example Problem



- Pushes the first feasible solution further into the search tree
- Still covers all possible permutations of value assignments to variables
- Still a valid CSP



Compare Three Backtracking Algortihms



1.) Chronological Backtracking

2.) Conflict-Directed Backjumping

3.) Dynamic Backtracking

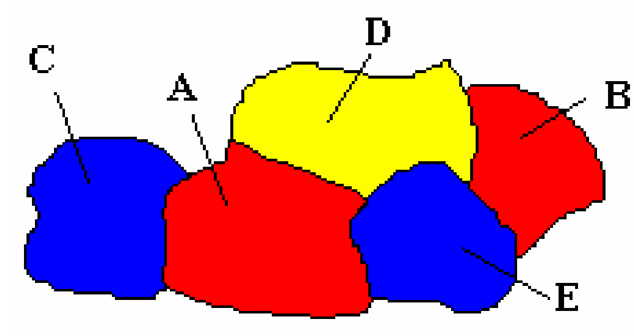
Sneak Preview of the Solution

Solve map example using:

1.) Chronological Backtracking

2.) Conflict-Directed Backjumping

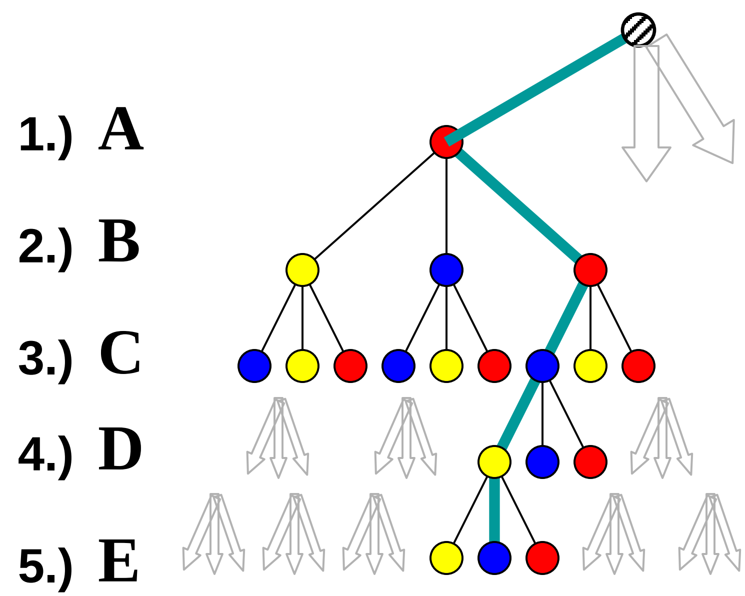
3.) Dynamic Backtracking



Note:

This is what the solution will look like each time.

We will compare the # of nodes expanded (i.e. regions colored) until the first solution is found



1.) Chronological Backtracking

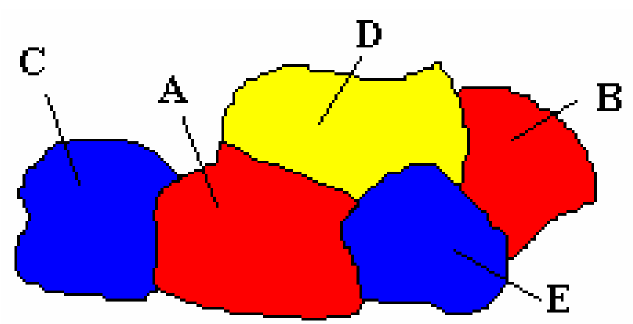
Chronological_Backtrack()

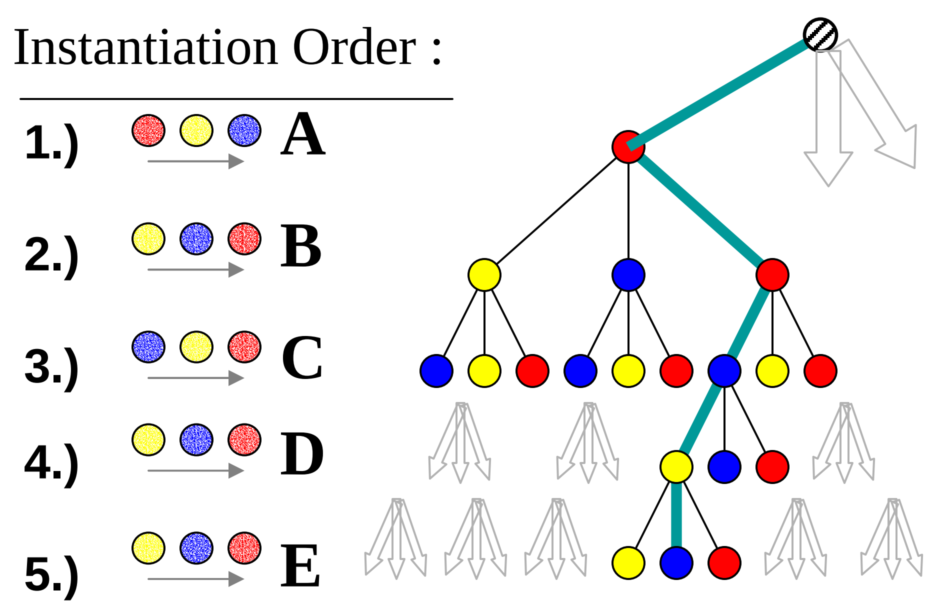
1.) Set P = {null} (P is the partial solution to the CSP)
Set Vi = {1} (start with first variable in instantiation order)

2.) If P = solution, return Success. If Vi = 0 return Failure
Else if P = Consistent,
set (Vi) to the next variable in instantiation order and assign it's next domain color (c).
Else if P = Inconsistent, remove (c) from domain of (Vi) and continue

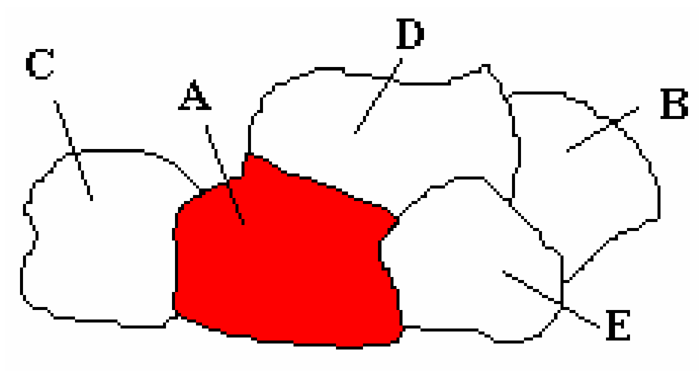
3.) While domain of (Vi) is not empty, choose the next domain color (c) and return to step 2.

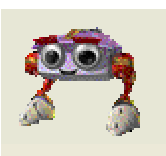
4.) If domain of (Vi) is empty (i.e. out of colors to try for (Vi)
Remove (Vi) from P, set Vi = Vi – 1, and return to step 3.





1.) Chronological Backtracking

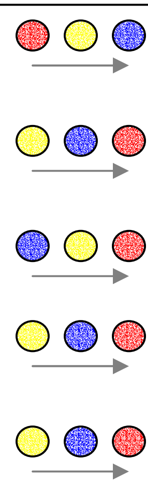




Notes:

Helpful notes will go here

Instantiation Order :



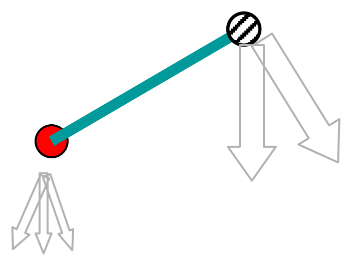
1.) A

2.) B

3.) C

4.) D

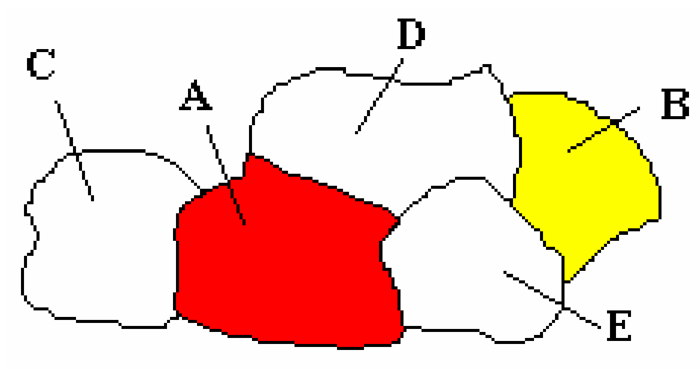
5.) E

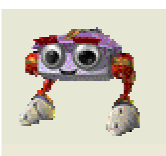


# of Nodes Expanded

1

1.) Chronological Backtracking

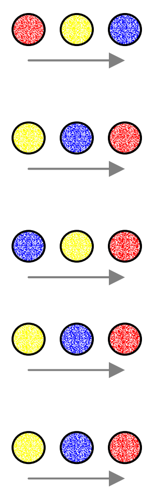




Notes:

Helpful notes will go here

Instantiation Order :



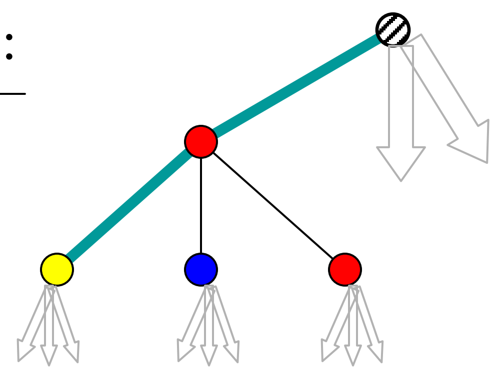
1.) A

2.) B

3.) C

4.) D

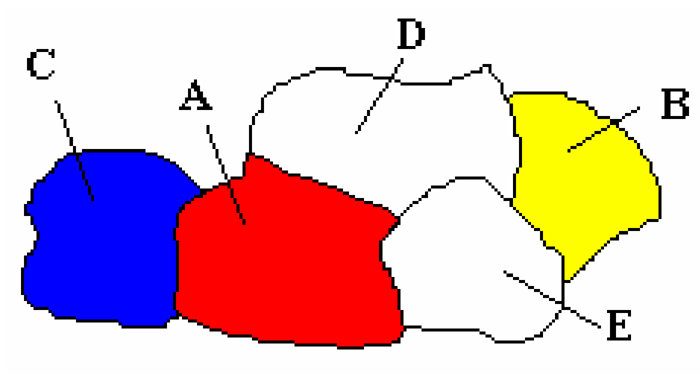
5.) E



# of Nodes Expanded

2

1.) Chronological Backtracking



Notes:

Instantiation Order :

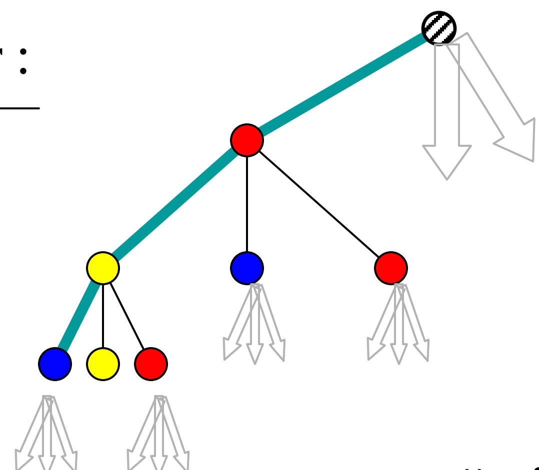
1.) A

2.) B

3.) C

4.) D

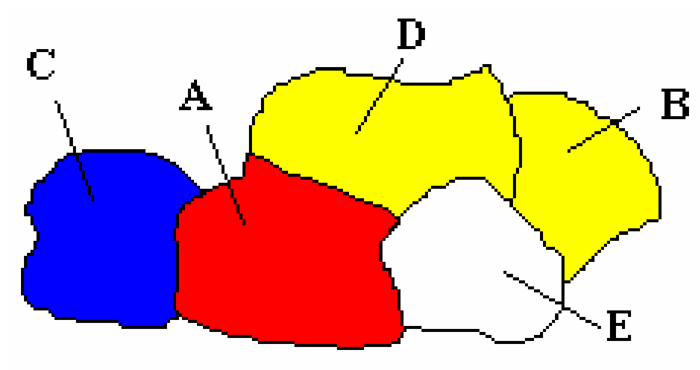
5.) E



# of Nodes Expanded

3

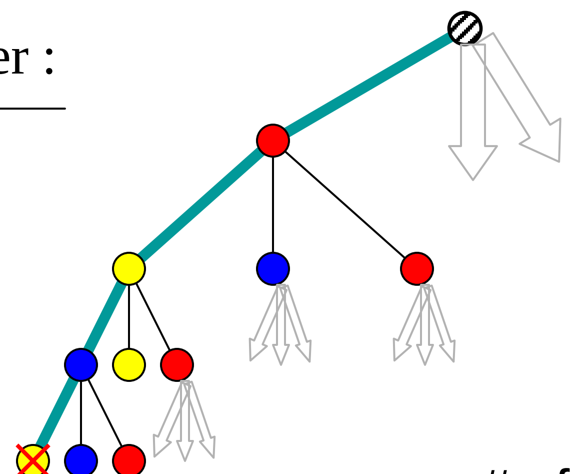
1.) Chronological Backtracking



Notes:

Instantiation Order :

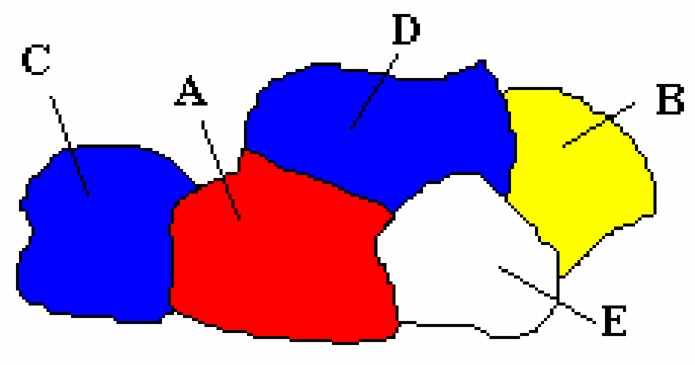
1.) A
2.) B
3.) C
4.) D
5.) E



# of Nodes Expanded

4

1.) Chronological Backtracking



Notes:

Instantiation Order :

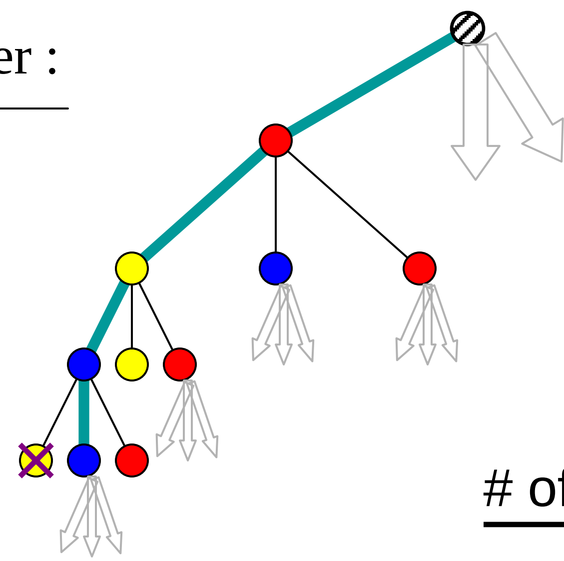
1.) A

2.) B

3.) C

4.) D

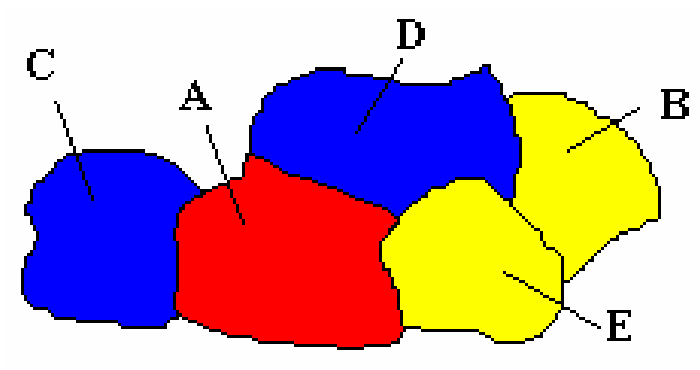
5.) E



# of Nodes Expanded

5

1.) Chronological Backtracking



Notes:

Instantiation Order :

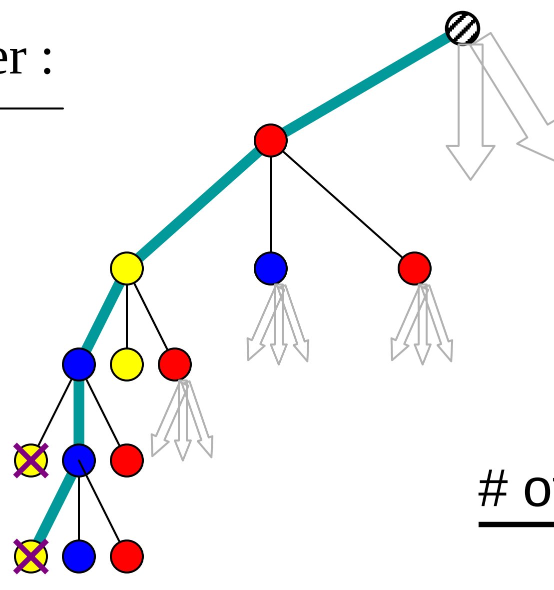
1.) A

2.) B

3.) C

4.) D

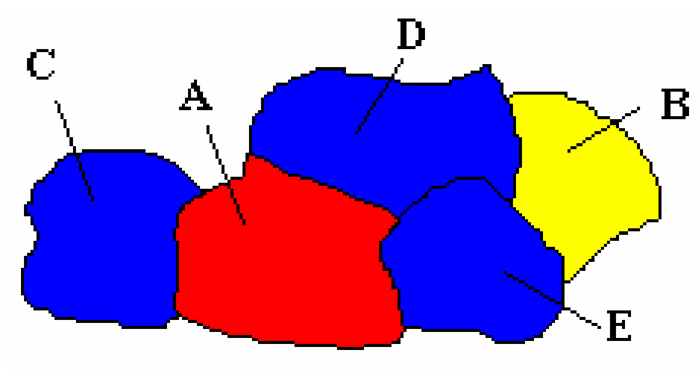
5.) E



# of Nodes Expanded

6

1.) Chronological Backtracking



Notes:

Instantiation Order :

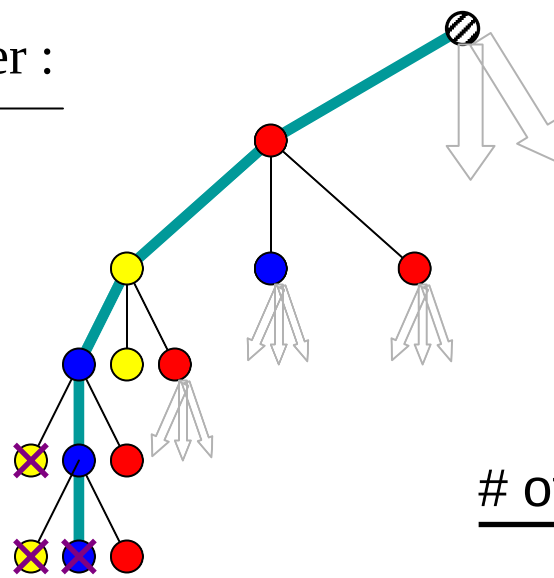
1.) A

2.) B

3.) C

4.) D

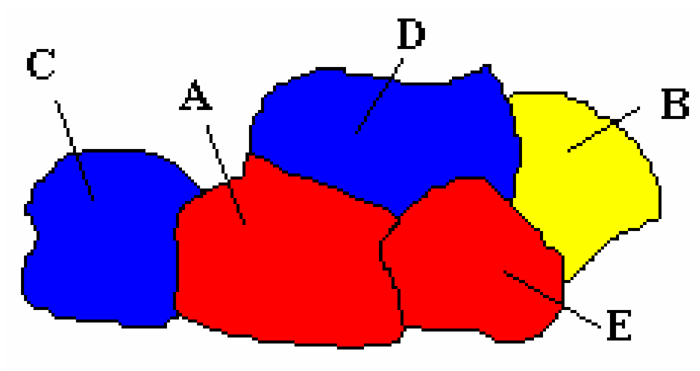
5.) E



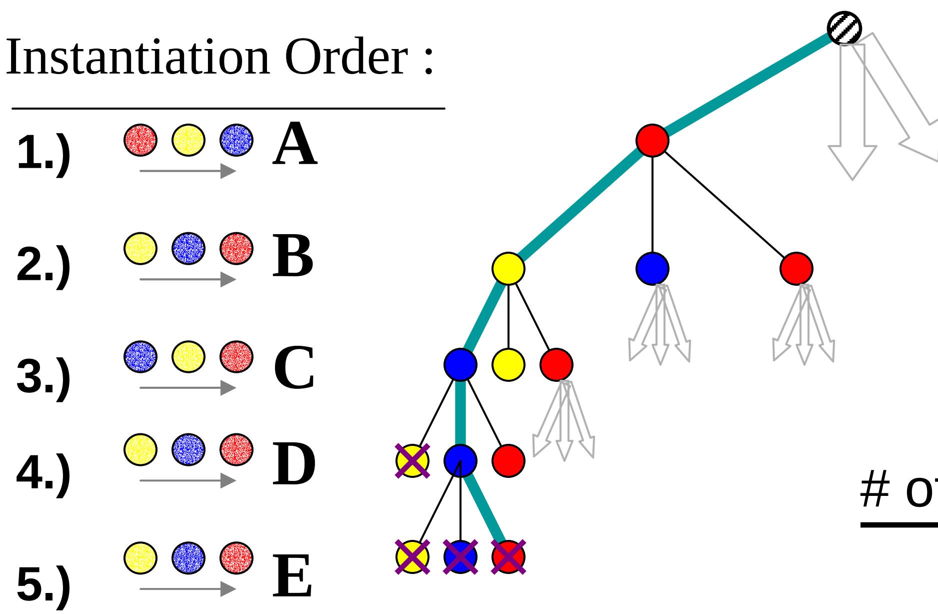
# of Nodes Expanded

7

1.) Chronological Backtracking



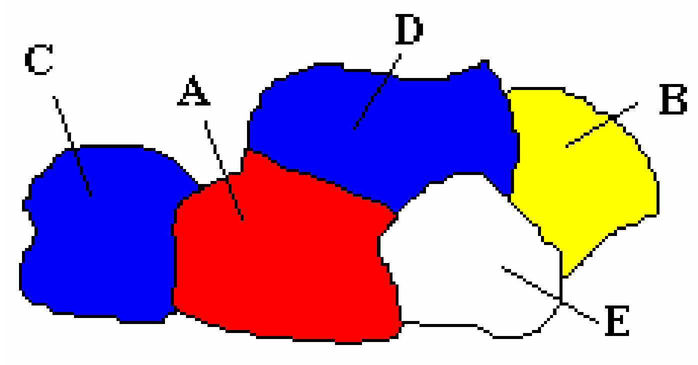
Notes:



# of Nodes Expanded

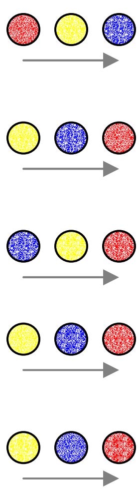
8

1.) Chronological Backtracking



Notes:

Instantiation Order :



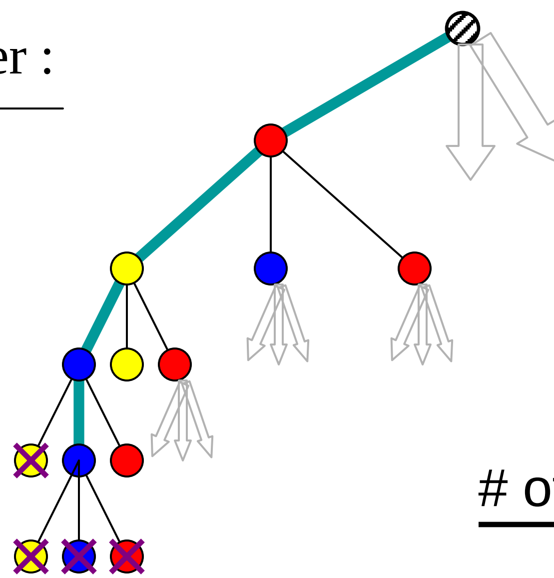
1.) A

2.) B

3.) C

4.) D

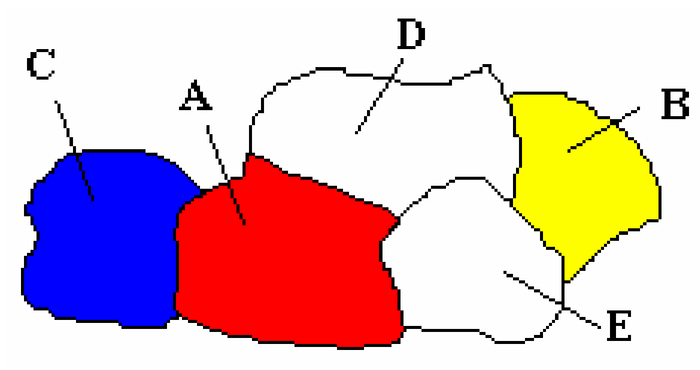
5.) E



# of Nodes Expanded

8

1.) Chronological Backtracking



Notes:

Instantiation Order :

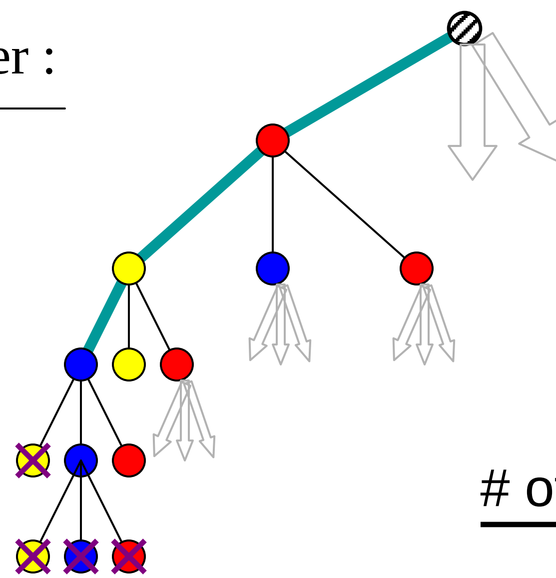
1.) A

2.) B

3.) C

4.) D

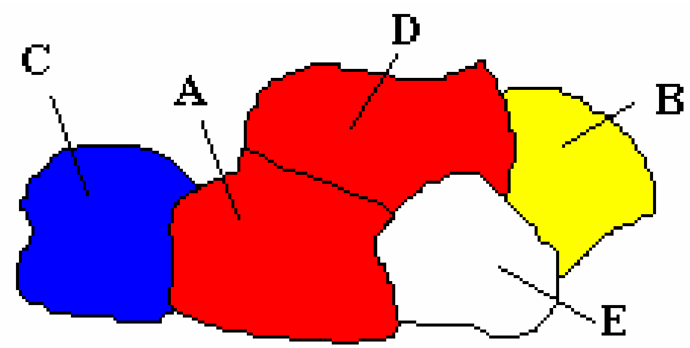
5.) E



# of Nodes Expanded

8

1.) Chronological Backtracking



Notes:

Instantiation Order :

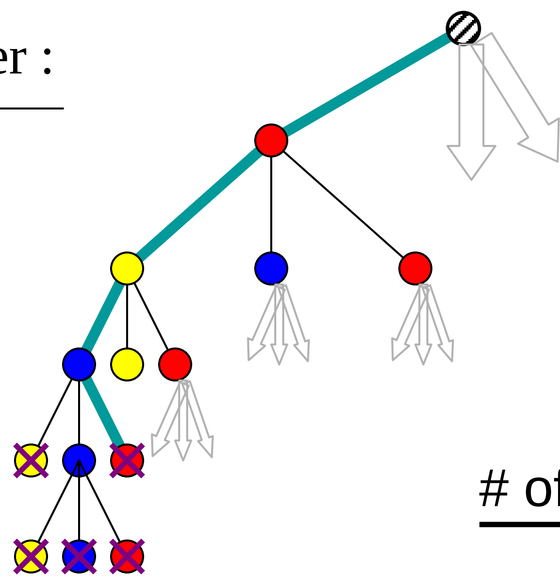
1.) A

2.) B

3.) C

4.) D

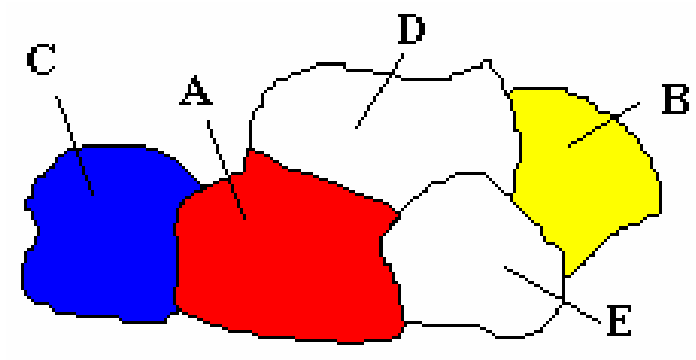
5.) E



# of Nodes Expanded

9

1.) Chronological Backtracking



Notes:

Instantiation Order :

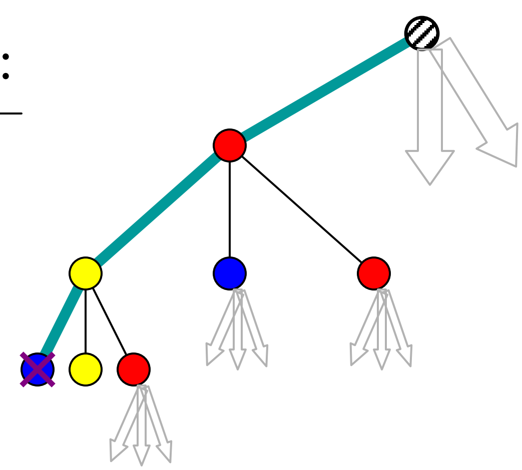
1.) A

2.) B

3.) C

4.) D

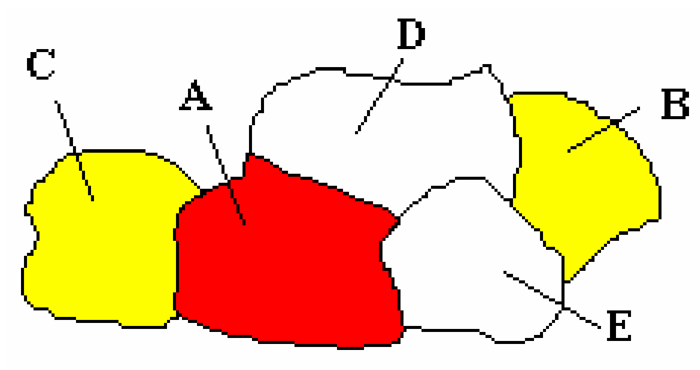
5.) E



# of Nodes Expanded

9

1.) Chronological Backtracking



Notes:

Instantiation Order :

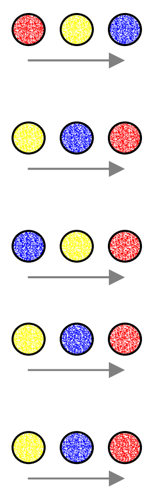
1.) A

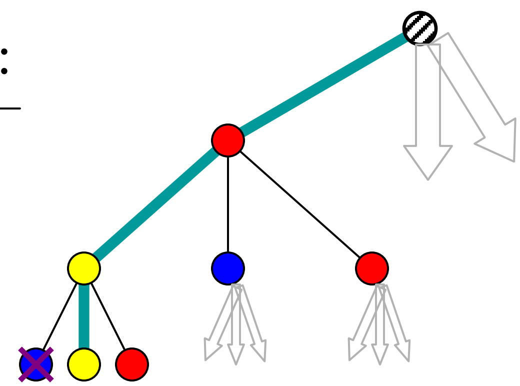
2.) B

3.) C

4.) D

5.) E

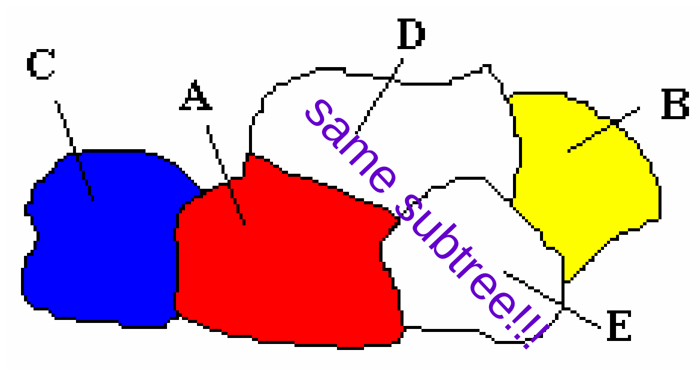




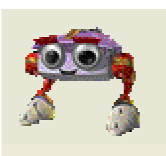
# of Nodes Expanded

10

1.) Chronological Backtracking



BT searches same subtree again!!!



Notes:

Chronological backtracking doesn't notice this is the same subtree, and still searches it.

(a.k.a. thrashing)

Instantiation Order :

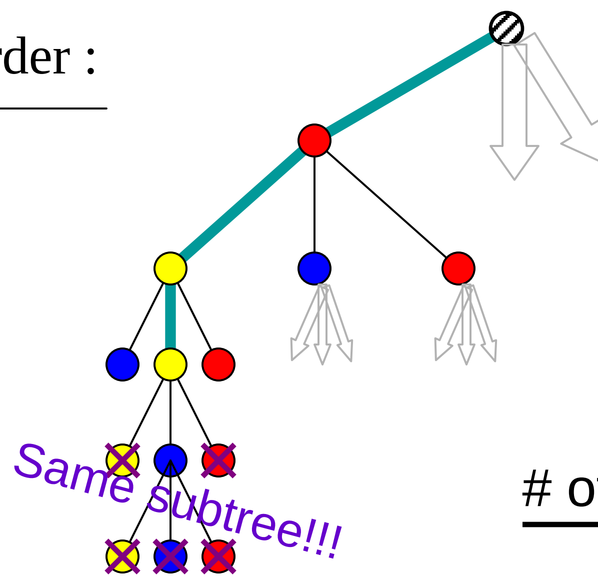
1.) A

2.) B

3.) C

4.) D

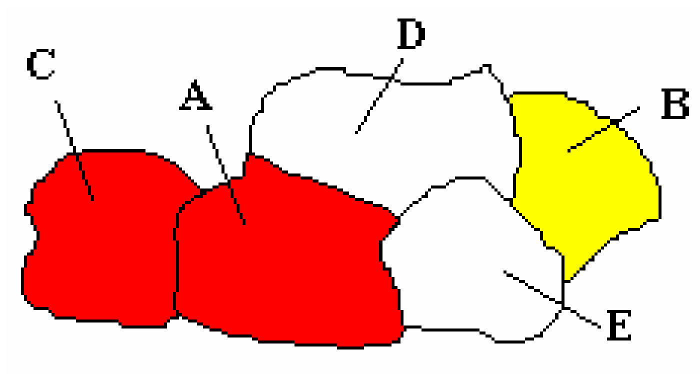
5.) E



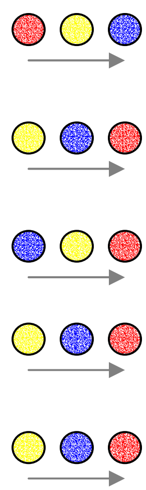
# of Nodes Expanded

15

1.) Chronological Backtracking



Instantiation Order :



1.) A

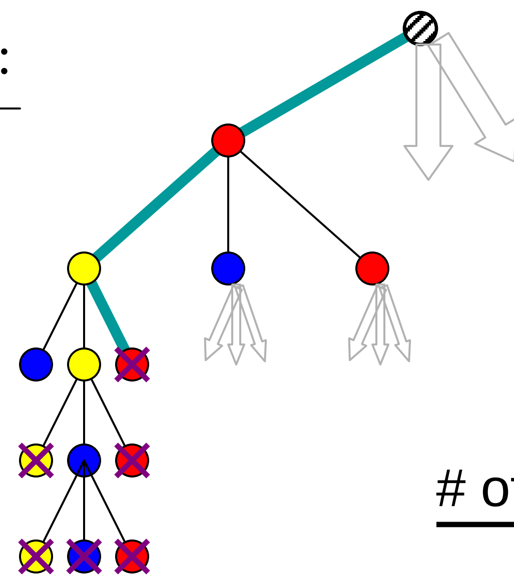
2.) B

3.) C

4.) D

5.) E

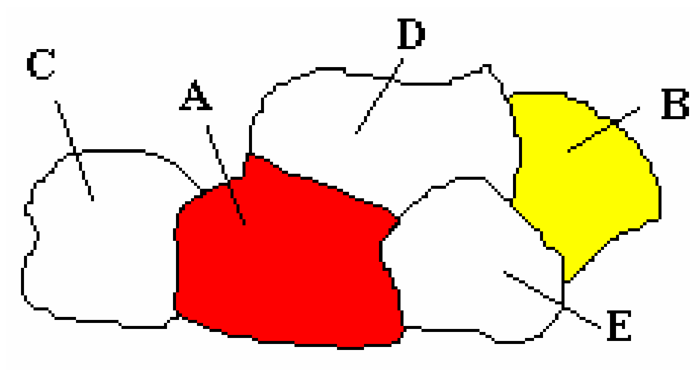
Notes:



# of Nodes Expanded

16

1.) Chronological Backtracking



Notes:

Instantiation Order :

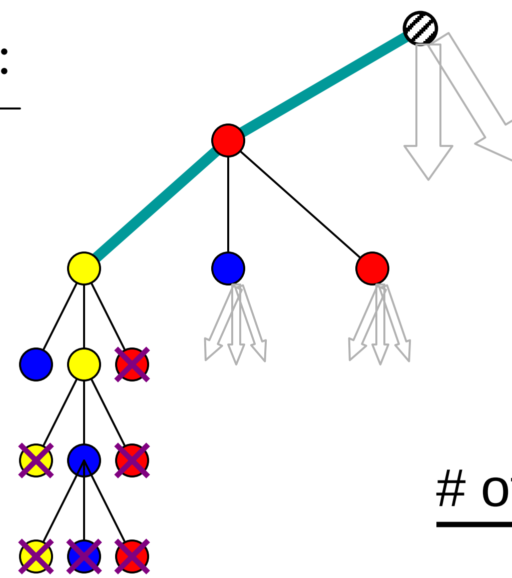
1.) A

2.) B

3.) C

4.) D

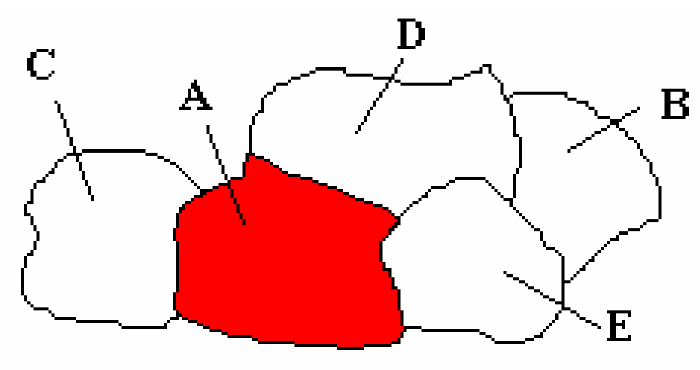
5.) E



# of Nodes Expanded

16

1.) Chronological Backtracking



Notes:

Instantiation Order :

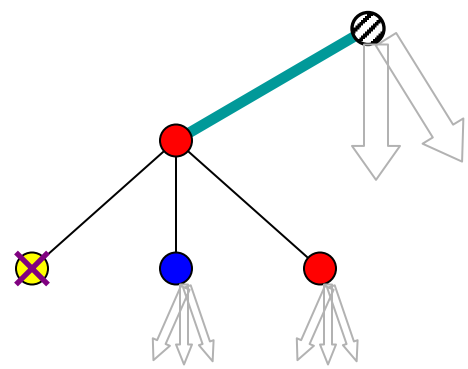
1.) A

2.) B

3.) C

4.) D

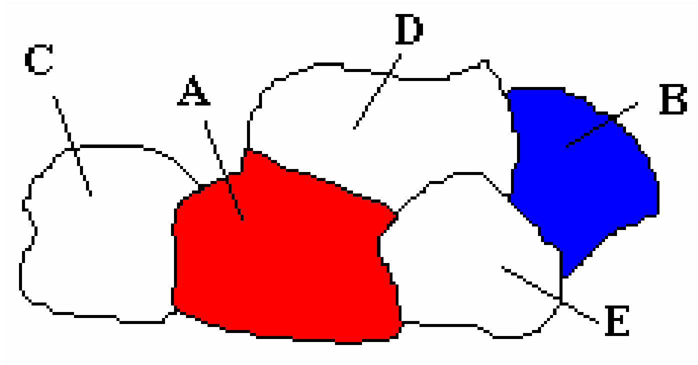
5.) E



# of Nodes Expanded

16

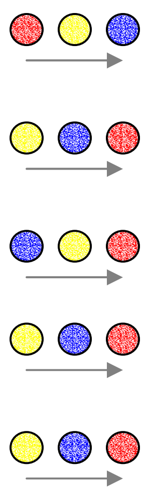
1.) Chronological Backtracking



BT searches wrong subtree again!!!

Notes:

Instantiation Order :



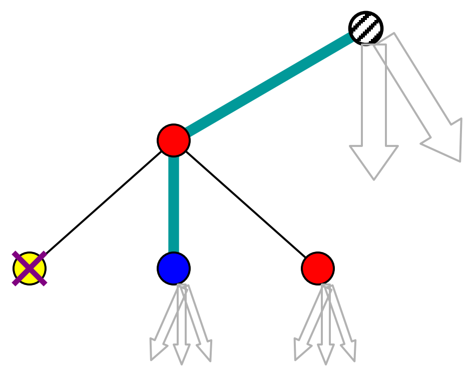
1.) A

2.) B

3.) C

4.) D

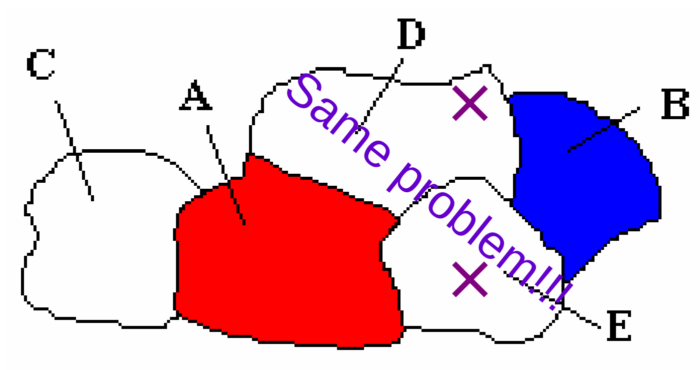
5.) E



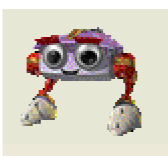
# of Nodes Expanded

16

1.) Chronological Backtracking



BT searches wrong subtree again!!!



Notes:

Search is repeating the same problem.

No solution is possible.

Instantiation Order :

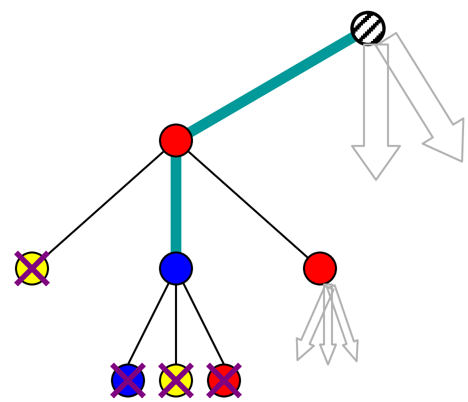
1.) A

2.) B

3.) C

4.) D

5.) E

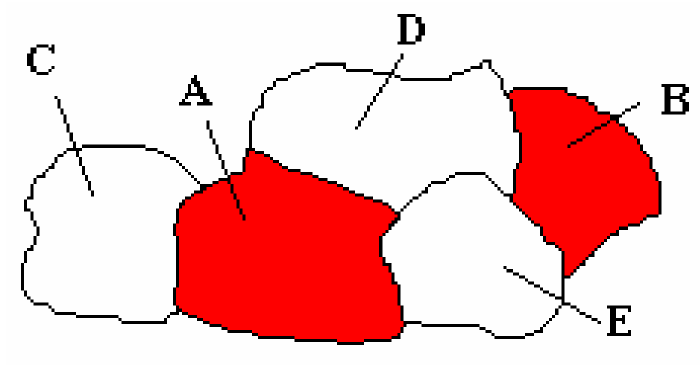


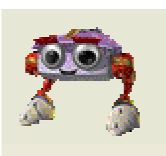
Cost = 16

# of Nodes Expanded

32

1.) Chronological Backtracking

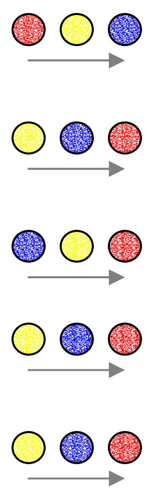




Notes:

Now we're getting somewhere

Instantiation Order :



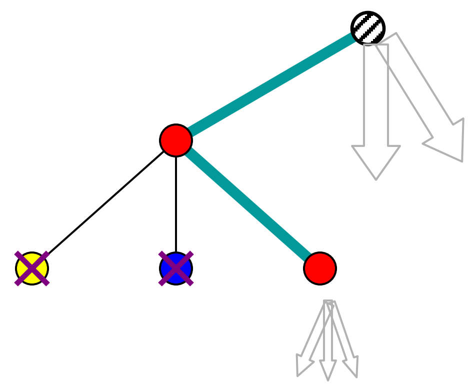
1.) A

2.) B

3.) C

4.) D

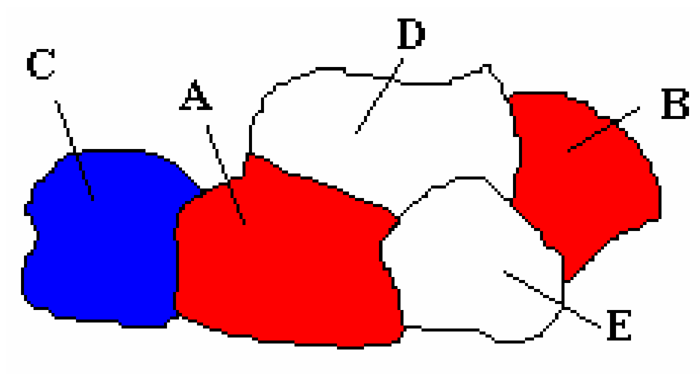
5.) E

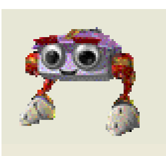


# of Nodes Expanded

33

1.) Chronological Backtracking

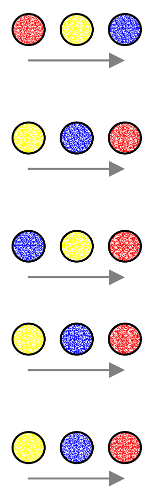




Notes:

Now we’re getting somewhere

Instantiation Order :



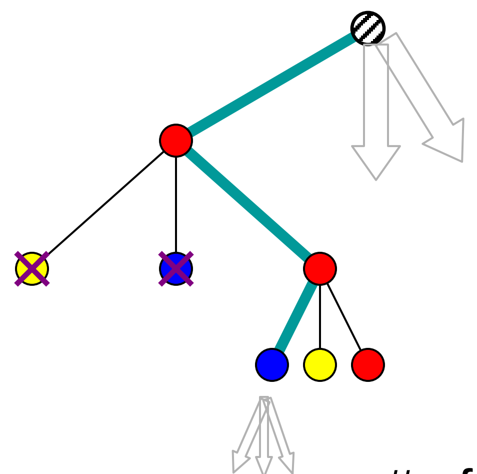
1.) A

2.) B

3.) C

4.) D

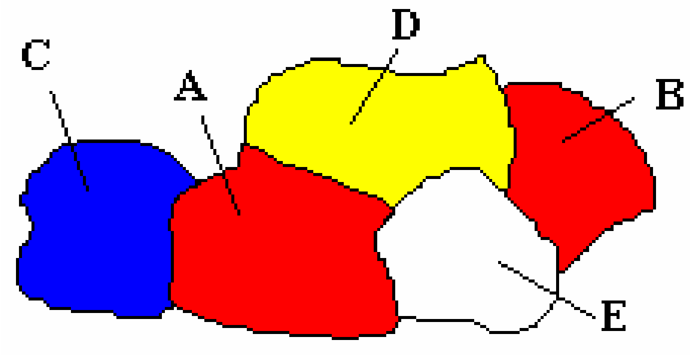
5.) E

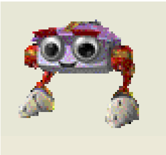


# of Nodes Expanded

34

1.) Chronological Backtracking





Notes:

Now we're getting somewhere

Instantiation Order :

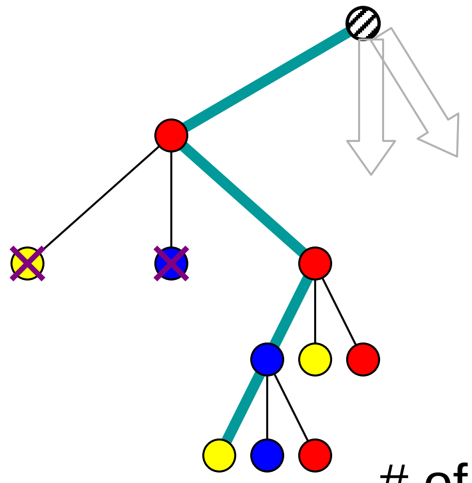
1.) A

2.) B

3.) C

4.) D

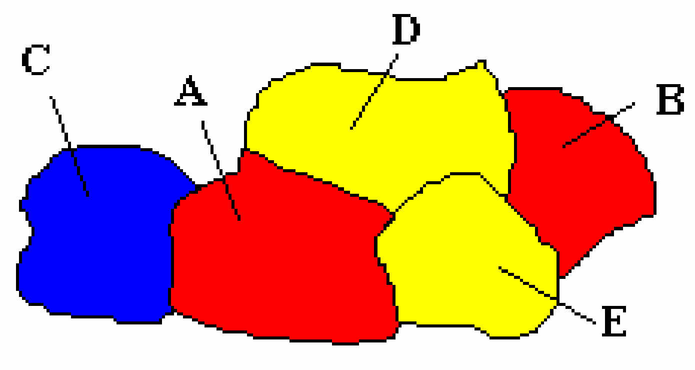
5.) E

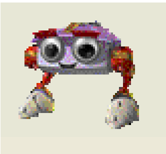


# of Nodes Expanded

35

1.) Chronological Backtracking

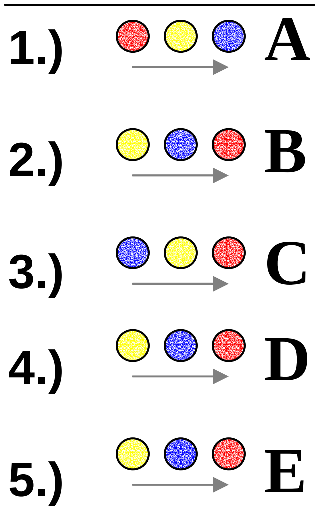


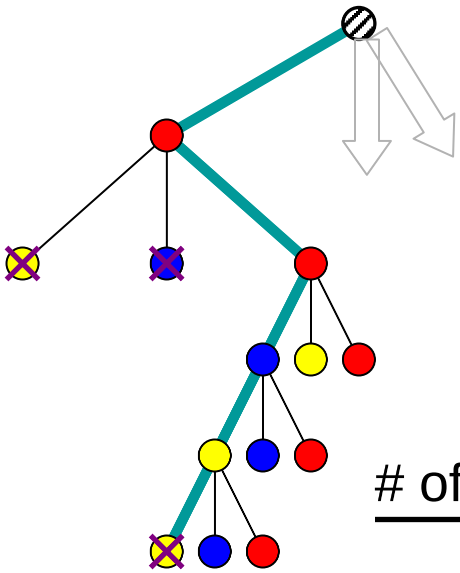


Notes:

Now we're getting somewhere

Instantiation Order :

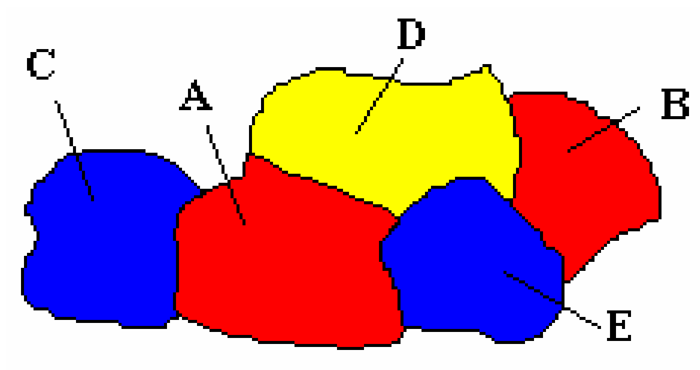


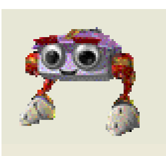


# of Nodes Expanded

36

1.) Chronological Backtracking





Notes:

Solution Found!!

# of Nodes Checked

= 37

Instantiation Order :

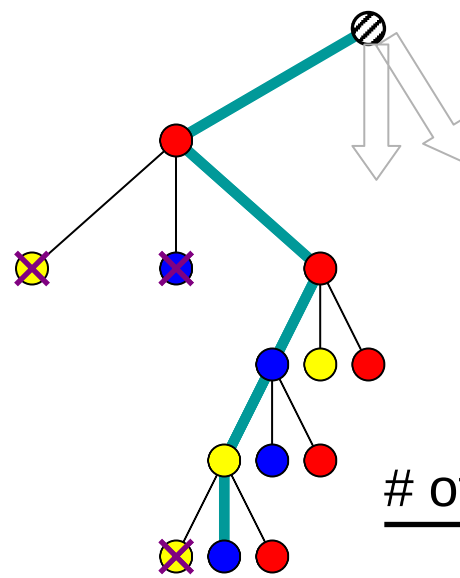
1.) A

2.) B

3.) C

4.) D

5.) E



# of Nodes Expanded

37

2.) Conflict-Directed Backjumping

2.) Conflict-Directed Backjumping

Conflict-Directed Backjumping()

1.) Set P = {null} and **Set E = {null}**

2.) If P = solution, return Success. If Vi = 0 return Failure.

Else if P = Consistent,

set next as (Vi) and assign next color (c), and goto step 2.

Else if P = Inconsistent,

remove (c) from domain of (Vi) and continue

3.) While domain of (Vi) is not empty

choose the next domain color (c) and goto step 2.

4.) If domain of (Vi) is empty

add (Vi , c) to Ei

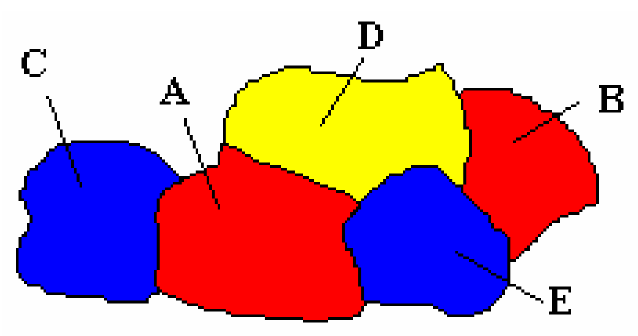
set Vi = most recent variable in Ei, and un-assign any variable later in the instantiation order

remove any (Ej) involving (Vi), return to step 3.

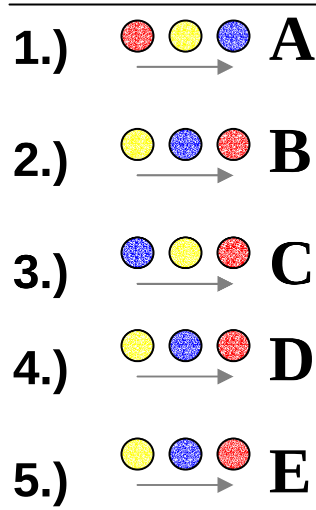
(E) Eliminating Explanations

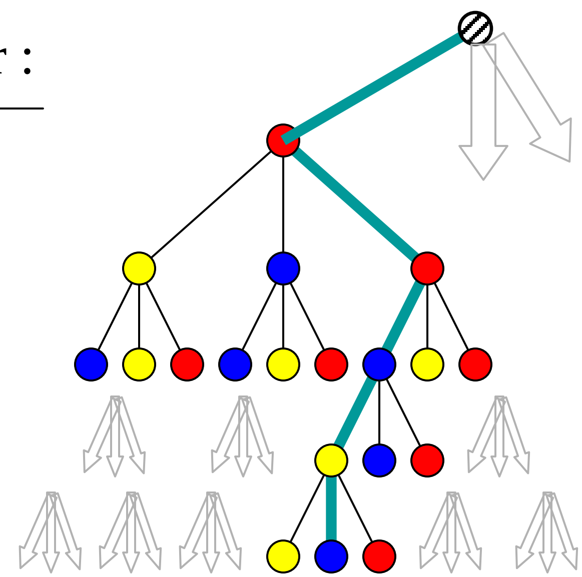
(A database of Conflicts)

there are many ways to store these conflicts

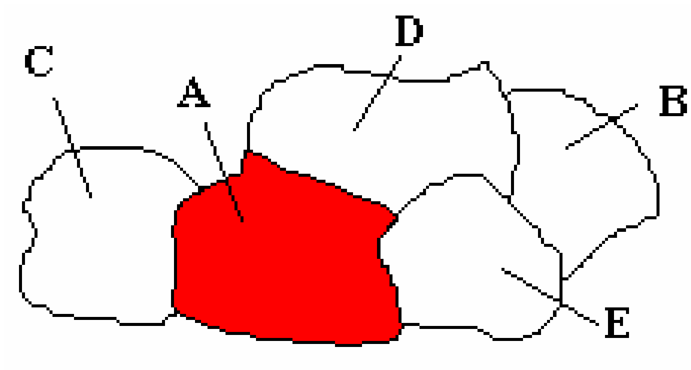


Instantiation Order :





2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B				
C				
D				
E				

Notes:

Helpful notes will go here.

Instantiation Order :

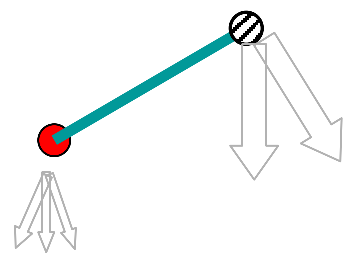
1.) **A**

2.) **B**

3.) **C**

4.) **D**

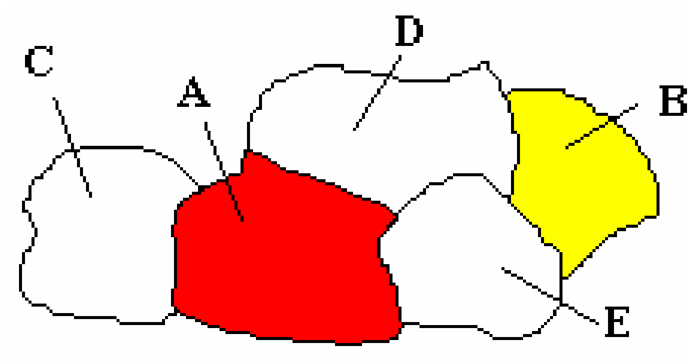
5.) **E**



# of Nodes Expanded

1

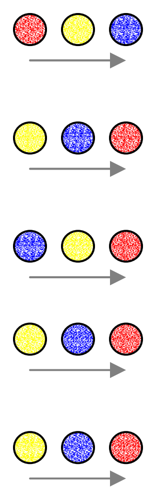
2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C				
D				
E				

Instantiation Order :



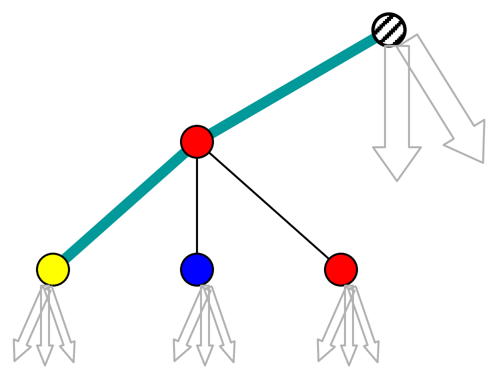
1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**

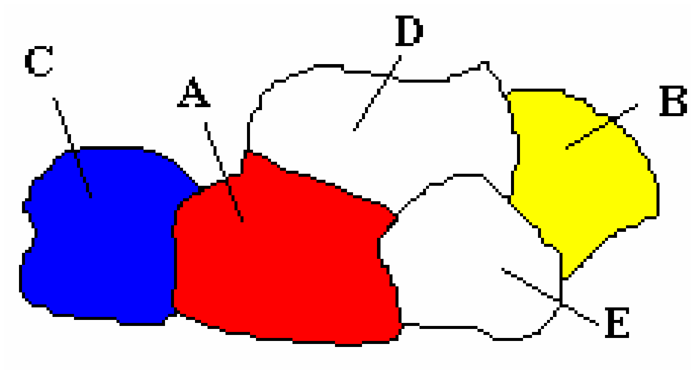


Notes:

# of Nodes Expanded

2

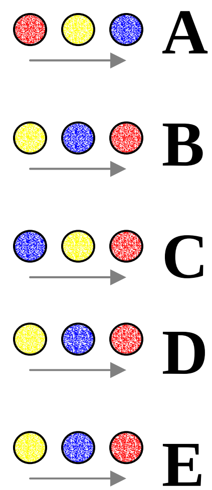
2.) Conflict-Directed Backjumping



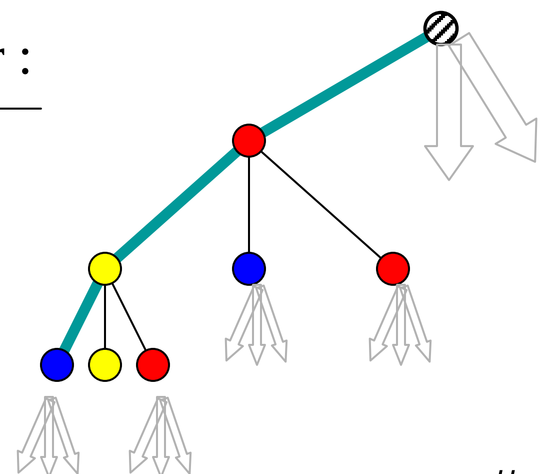
Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D				
E				

Instantiation Order :



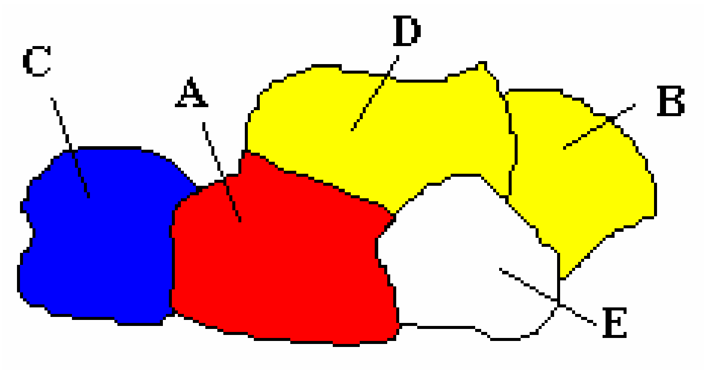
Notes:



# of Nodes Expanded

3

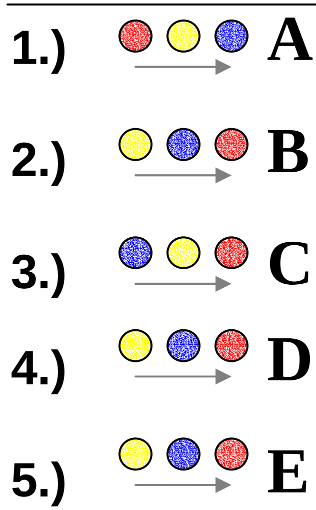
2.) Conflict-Directed Backjumping



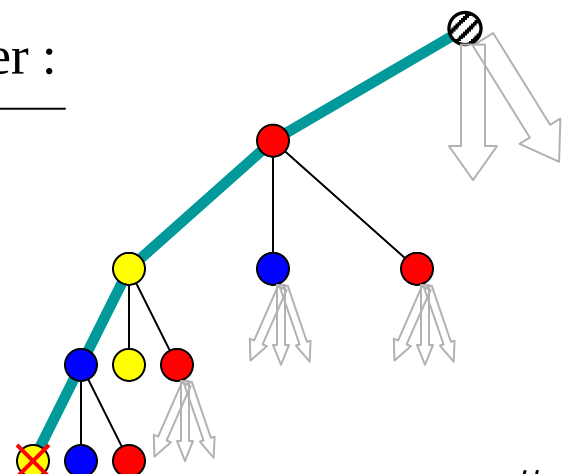
Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D	yellow		A,B	
E				

Instantiation Order :



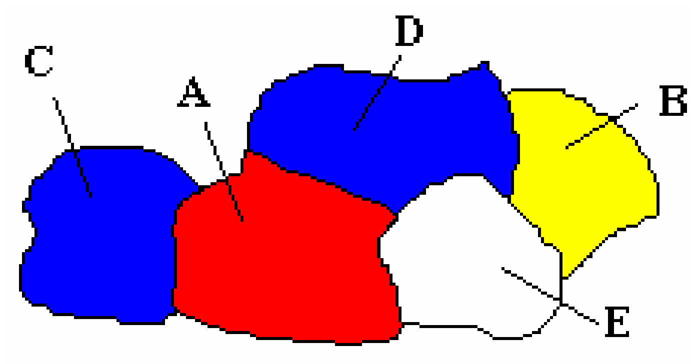
Notes:



# of Nodes Expanded

4

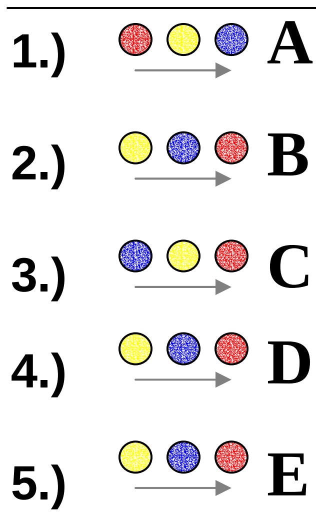
2.) Conflict-Directed Backjumping

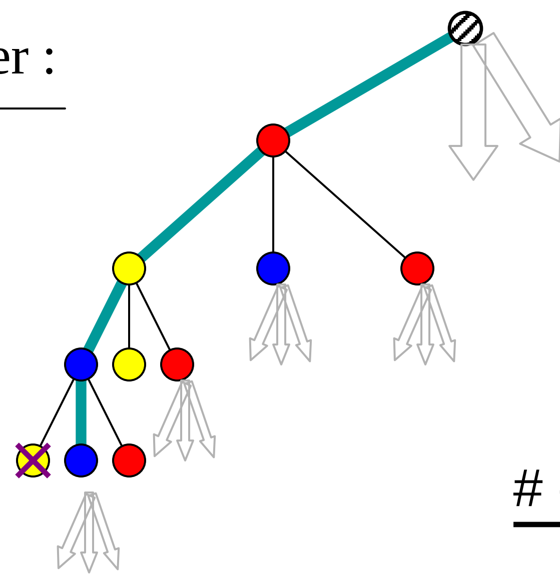


Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	
E				

Instantiation Order :



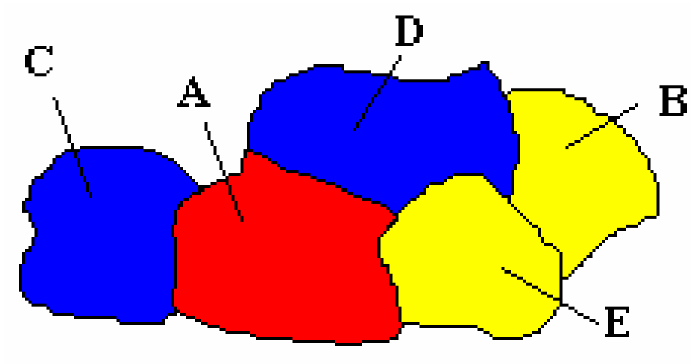


Notes:

# of Nodes Expanded

5

2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	
E	yellow		A,B,D	

Instantiation Order :

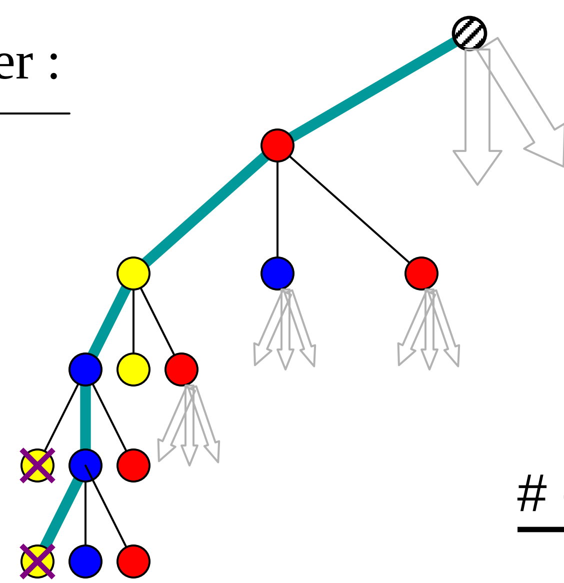
1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**

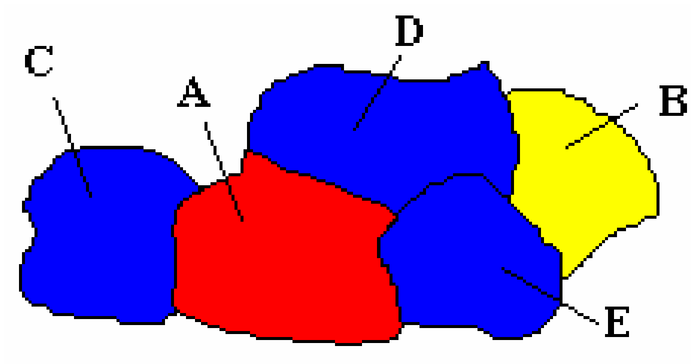


Notes:

# of Nodes Expanded

6

2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	
E	yellow		A,B,D	A,B,D

Instantiation Order :

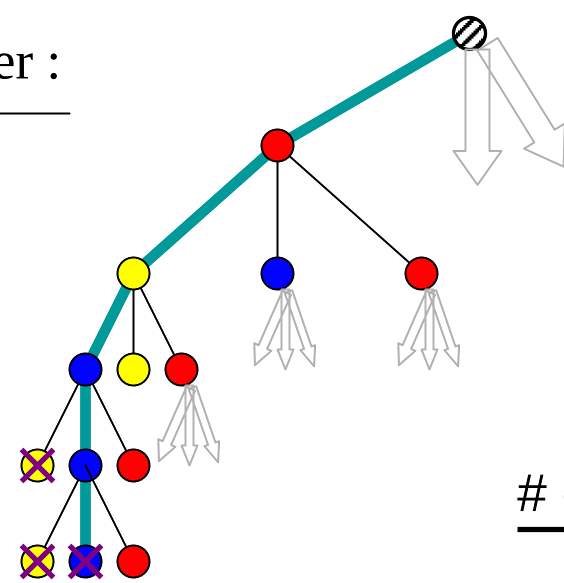
1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**

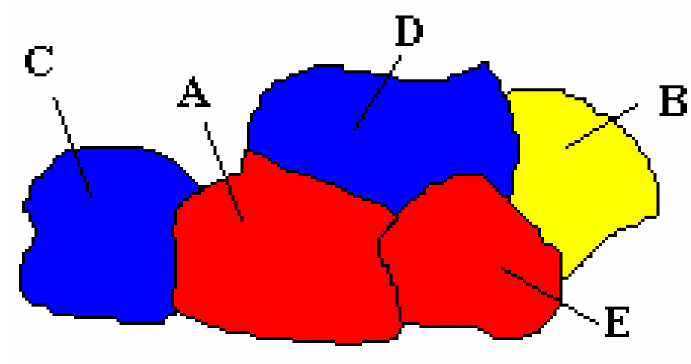


Notes:

# of Nodes Expanded

7

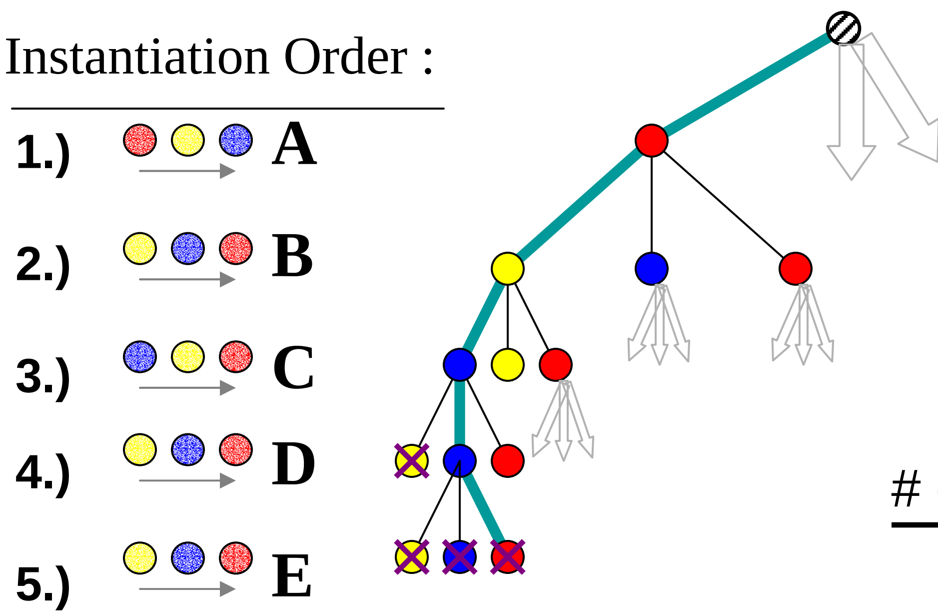
2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	
E	red	A,B,D	A,B,D	A,B,D

Instantiation Order :

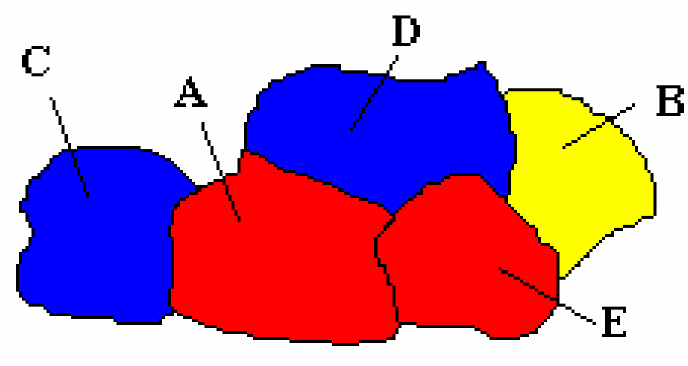


Notes:

# of Nodes Expanded

8

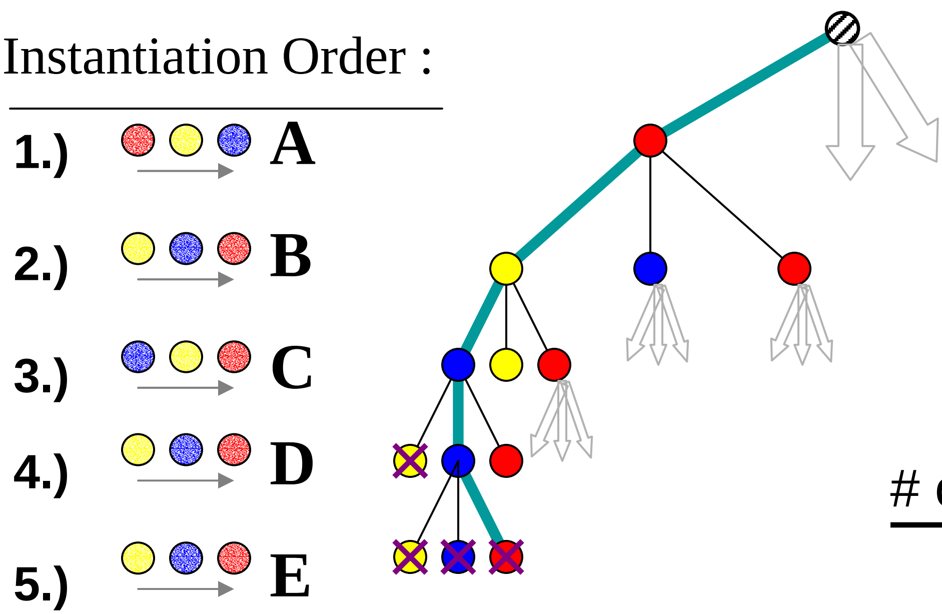
2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	A,B
E	~~red~~	A,B,D	A,B,D	A,B,D

Instantiation Order :

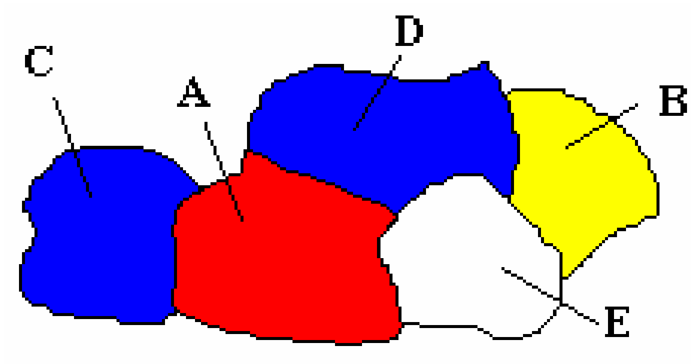


Notes:

# of Nodes Expanded

8

2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	yellow			
C	blue			
D	~~blue~~		A,B	A,B
E	~~red~~			

Notes:

Instantiation Order :

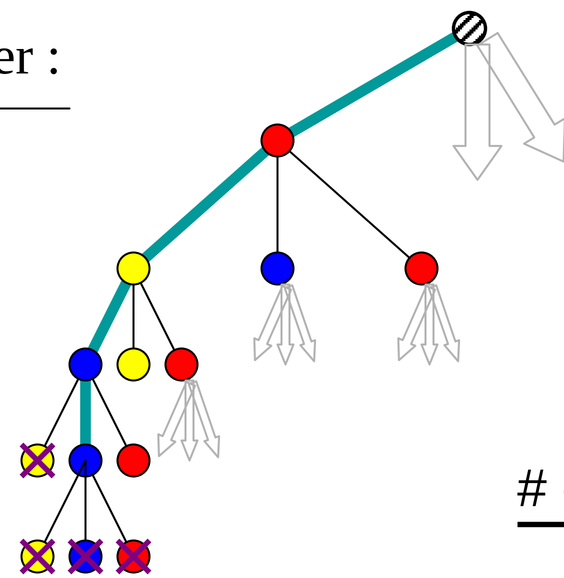
1.) A

2.) B

3.) C

4.) D

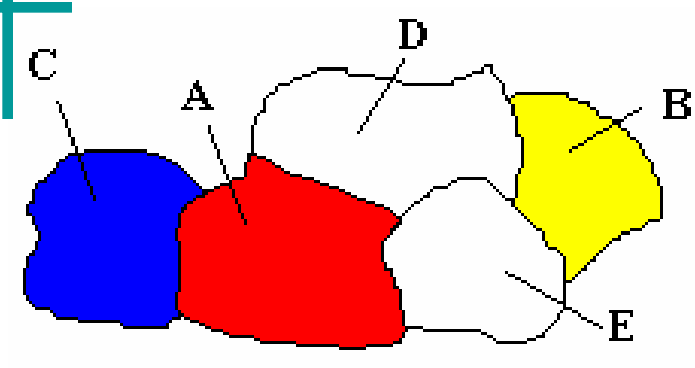
5.) E



# of Nodes Expanded

8

2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	yellow			
C	blue			
D	red		A,B	A,B
E				

Instantiation Order :

1.) **A**

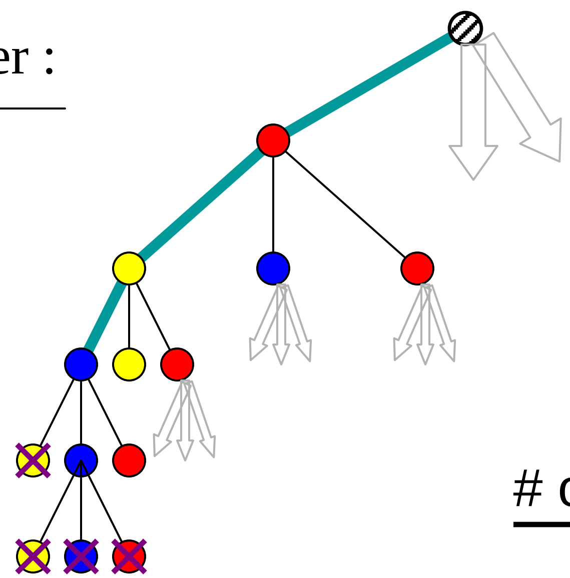
2.) **B**

3.) **C**

4.) **D**

5.) **E**

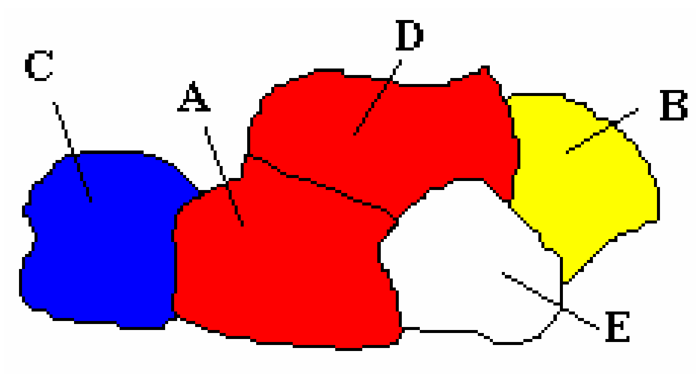
Notes:



# of Nodes Expanded

8

2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	~~yellow~~			
C	blue			
D	~~red~~	A,B	A,B	A,B
E				

Notes:

Instantiation Order :

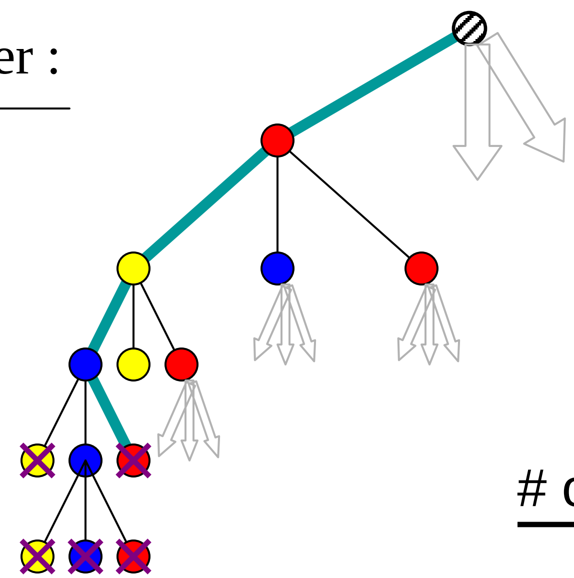
1.) A

2.) B

3.) C

4.) D

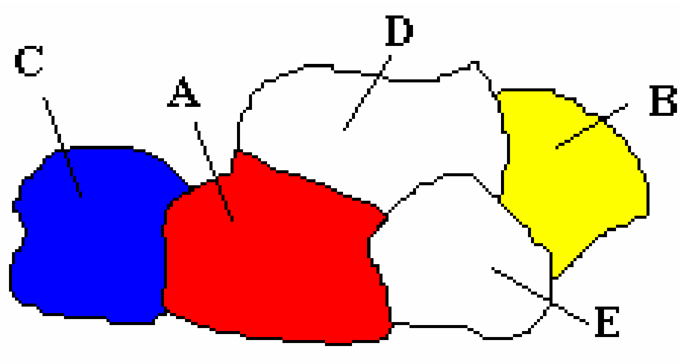
5.) E



# of Nodes Expanded

9

2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	~~yellow~~			
C	blue			
D		A,B	A,B	A,B
E				

Notes:

Conflict-Directed Backjumping knows to skip the circled nodes

Instantiation Order :

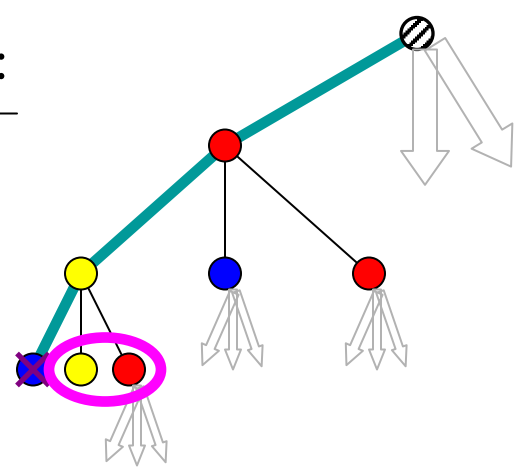
1.) A

2.) B

3.) C

4.) D

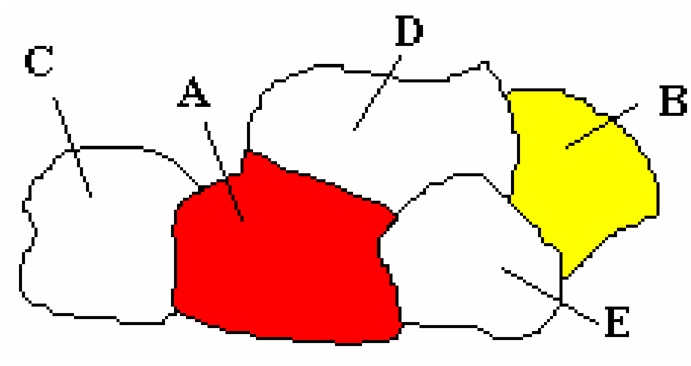
5.) E



# of Nodes Expanded

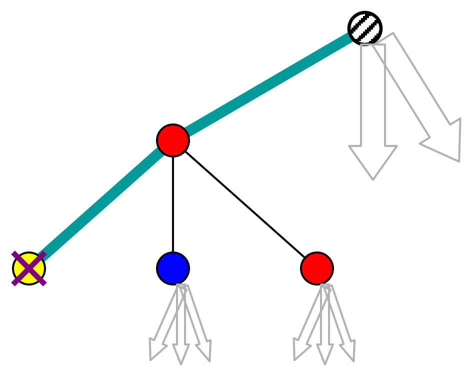
9

2.) Conflict-Directed Backjumping

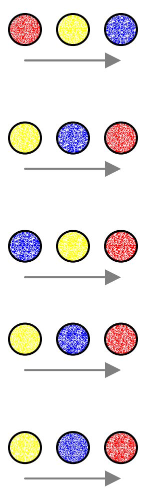


Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	~~yellow~~		A	
C				
D		A,B	A,B	A,B
E				



Instantiation Order :



1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**

Notes:

8 nodes expanded so far

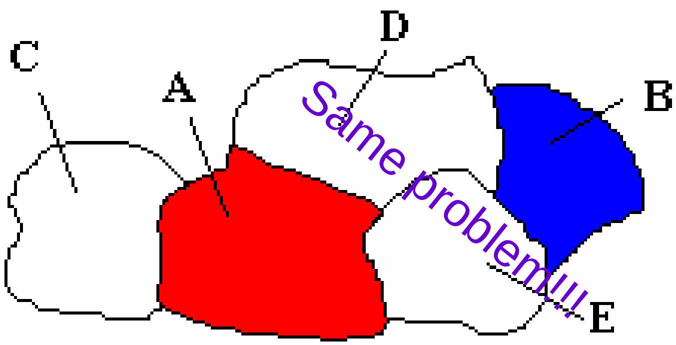
vs.

about 16 for chronological backtracking

# of Nodes Expanded

9

2.) Conflict-Directed Backjumping



Elimination Explanations:

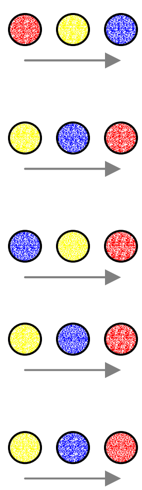
Region	Color	Red	yellow	blue
A	red			
B	~~blue~~		A	A
C				
D				
E				

Notes:

Will explore this tree even though no chance of success

Cost = 9 nodes better than the 16 nodes during chronological backtracking.

Instantiation Order :



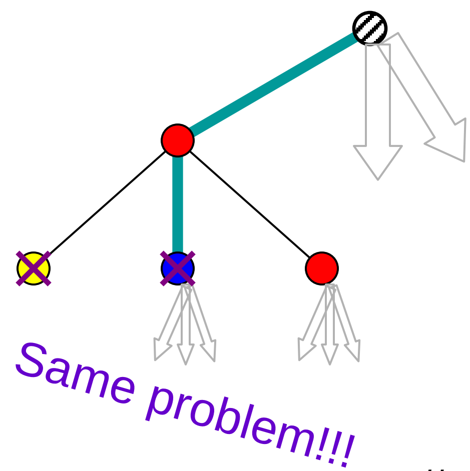
1.) A

2.) B

3.) C

4.) D

5.) E

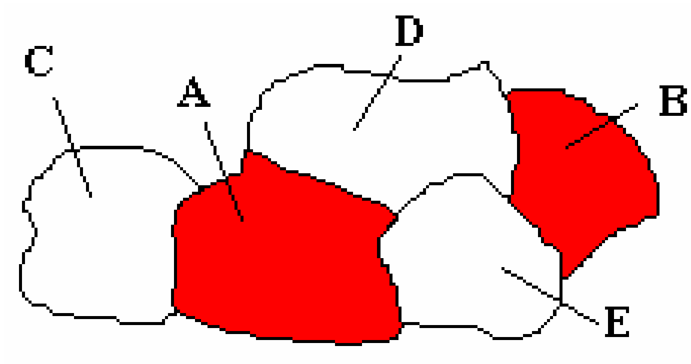


Cost = 9 nodes

# of Nodes Expanded

18

2.) Conflict-Directed Backjumping



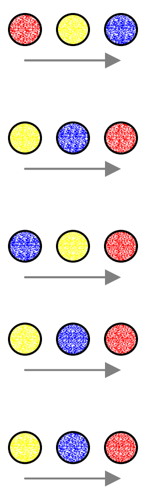
Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	red		A	A
C				
D				
E				

Notes:

Now we're on the right track

Instantiation Order :



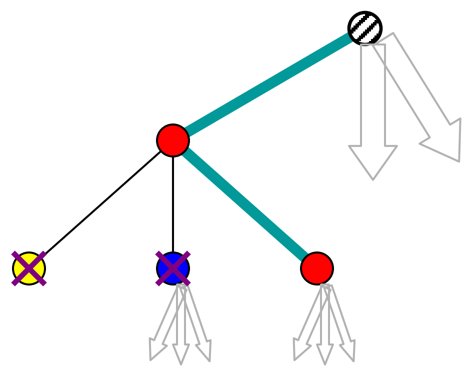
1.) **A**

2.) **B**

3.) **C**

4.) **D**

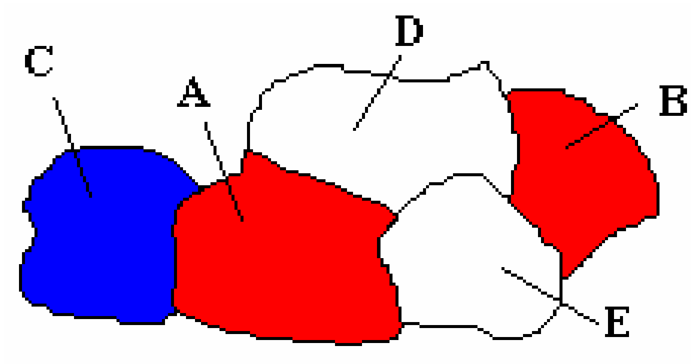
5.) **E**



# of Nodes Expanded

19

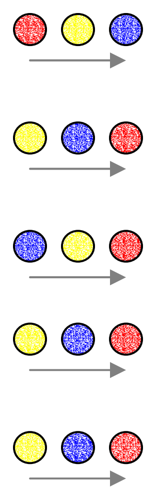
2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	red		A	A
C	blue			
D				
E				

Instantiation Order :



1.) **A**

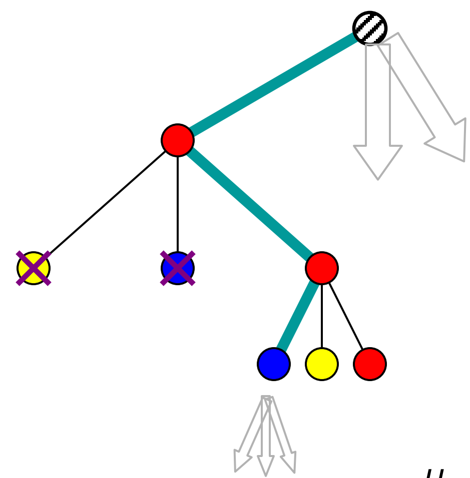
2.) **B**

3.) **C**

4.) **D**

5.) **E**

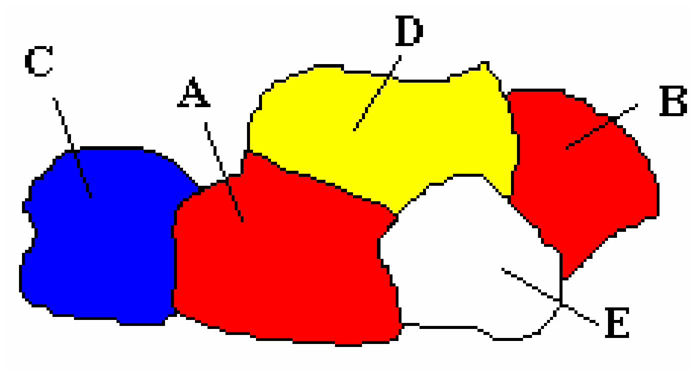
Notes:



# of Nodes Expanded

20

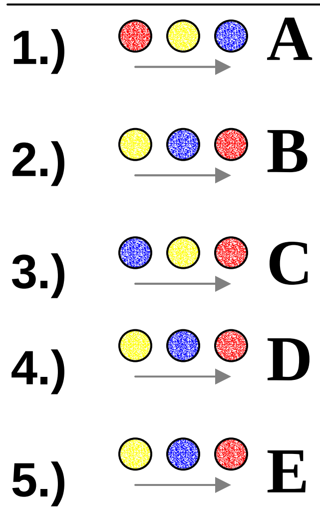
2.) Conflict-Directed Backjumping



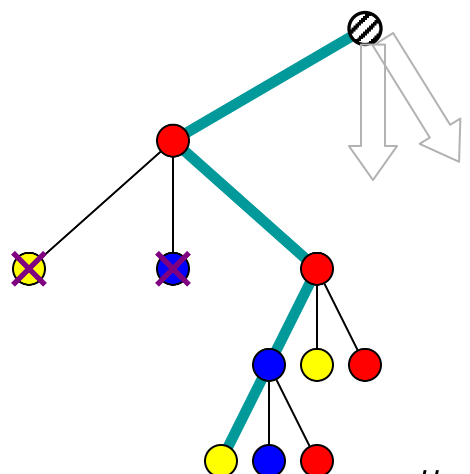
Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	red		A	A
C	blue			
D	yellow			
E				

Instantiation Order :



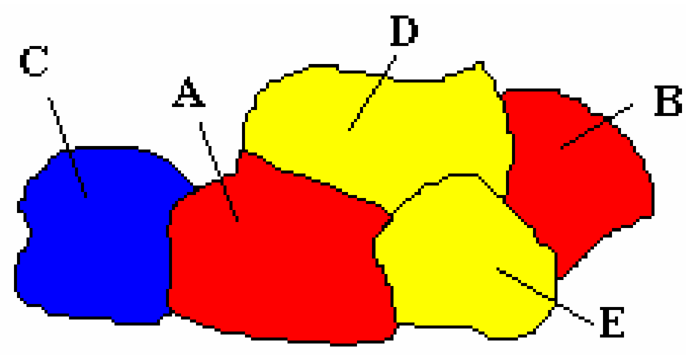
Notes:



# of Nodes Expanded

21

2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	red		A	A
C	blue			
D	yellow			
E	~~yellow~~		A,B,D	

Instantiation Order :

1.) **A**

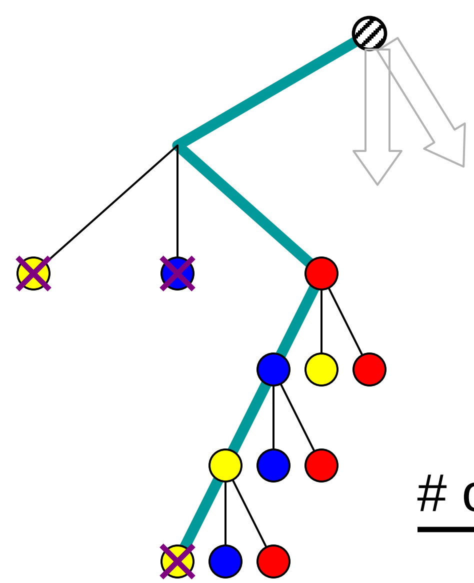
2.) **B**

3.) **C**

4.) **D**

5.) **E**

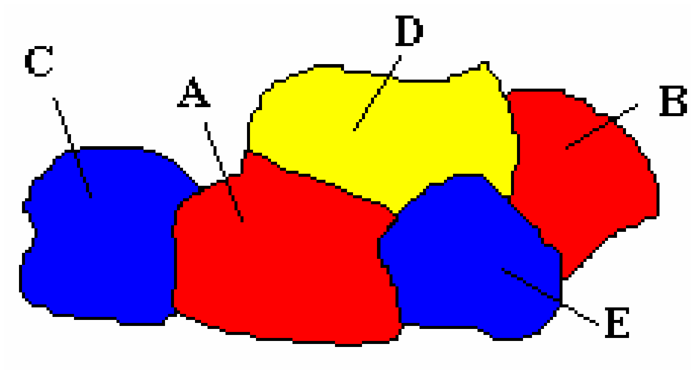
Notes:



# of Nodes Expanded

22

2.) Conflict-Directed Backjumping



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	red		A	A
C	blue			
D	yellow			
E	blue		A,B,D	

Notes:

Solution Found!!

Instantiation Order :

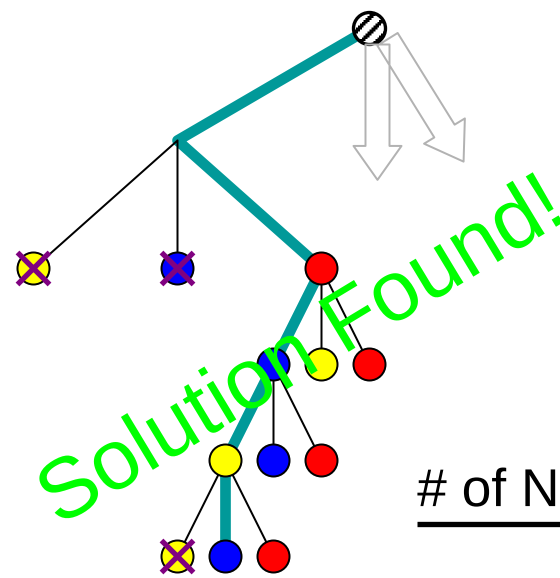
1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**



# of Nodes Expanded

23

3.) Dynamic Backtracking

3.) Dynamic Backtracking

Dynamic Backtracking()

1.) Set P = {null} and **Set E = {null}**

2.) If P = solution, return Success. If Vi = 0 return Failure.

Else if P = Consistent,

set next as (Vi) and assign next color (c), and goto step 2.

Else if P = Inconsistent,

remove (c) from domain of (Vi) and continue

3.) While domain of (Vi) is not empty

choose the next domain color (c) and goto step 2.

4.) If domain of (Vi) is empty

add (Vi , c) to Ei

Remove (Vi) from P

set Vi = ~~most recent variable in Ei~~ any variable not in P

remove any (Ej) involving (Vi), return to step 3.

Elimination Explanations:

Region	Color	red	yellow	blue
A				
B				
C				
D				
E				

Allows a dynamic Instantiation order!

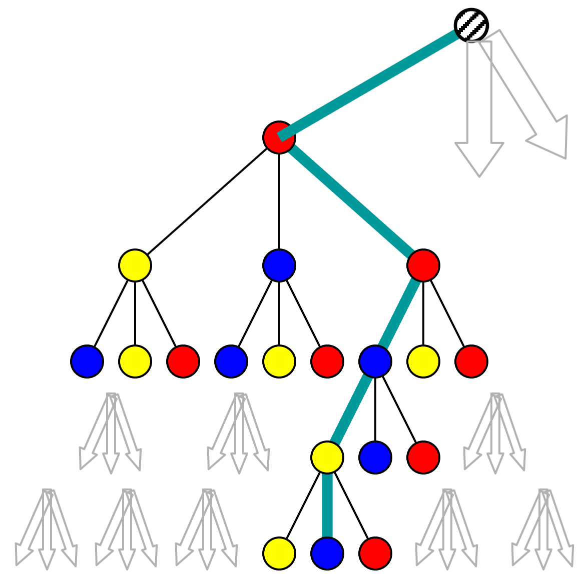
1.) A

2.) B

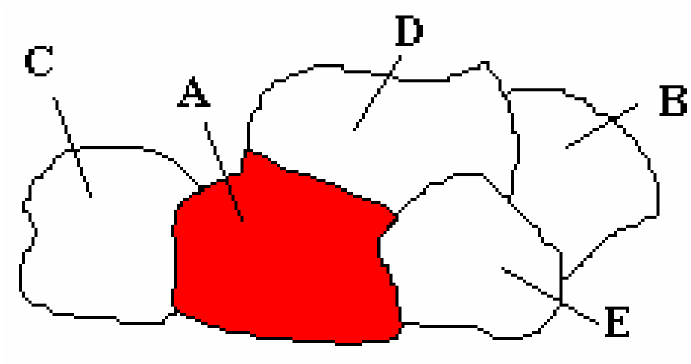
3.) C

4.) D

5.) E



3.) Dynamic Backtracking



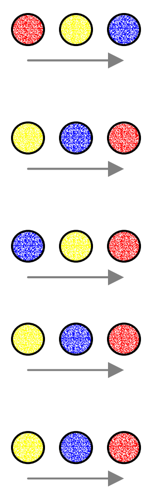
Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B				
C				
D				
E				

Notes:

Helpful notes will go here.

Instantiation Order :



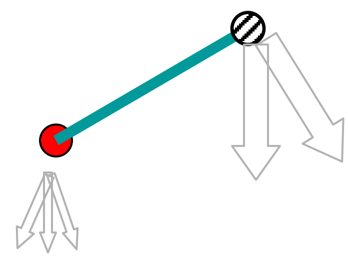
1.) A

2.) B

3.) C

4.) D

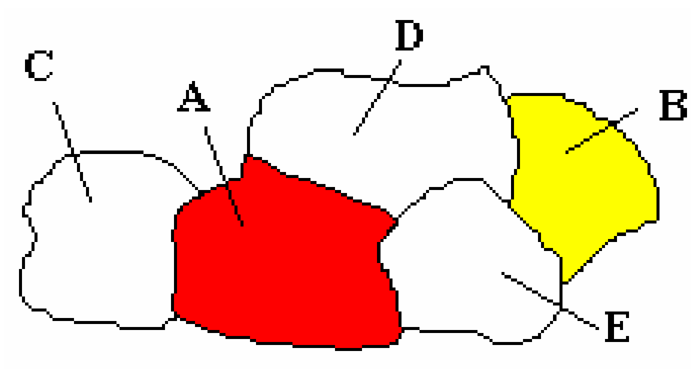
5.) E



# of Nodes Expanded

1

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C				
D				
E				

Instantiation Order :

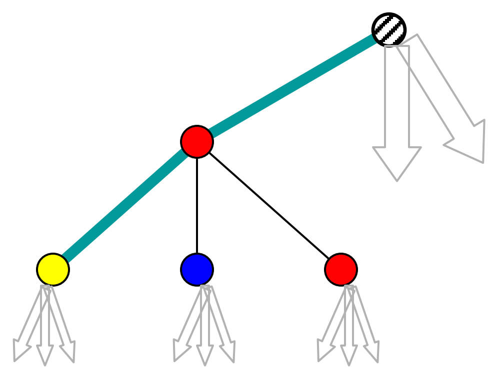
1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**

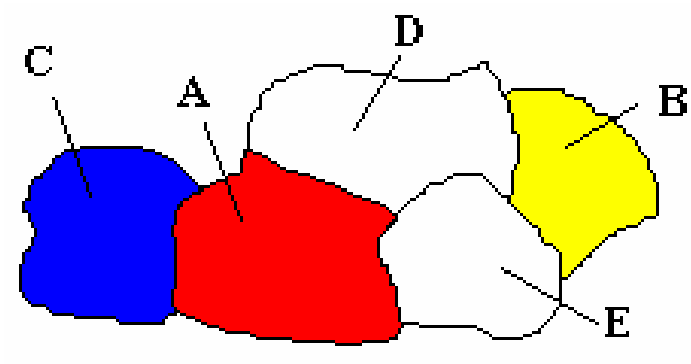


Notes:

# of Nodes Expanded

2

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D				
E				

Instantiation Order :

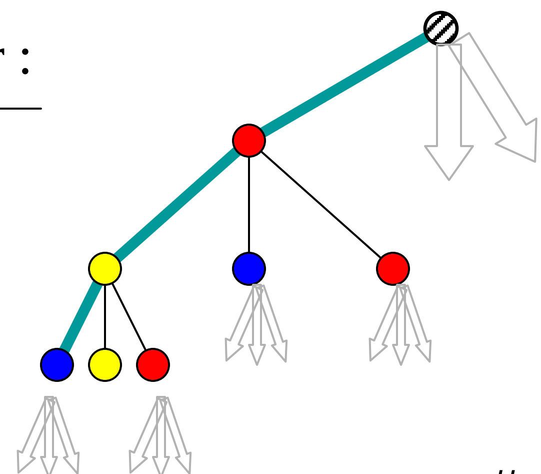
1.) A

2.) B

3.) C

4.) D

5.) E

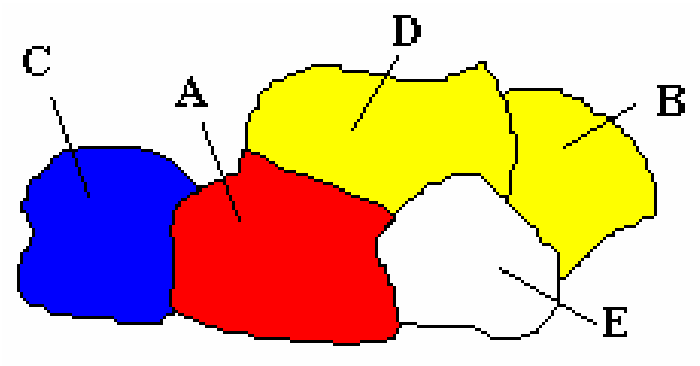


Notes:

# of Nodes Expanded

3

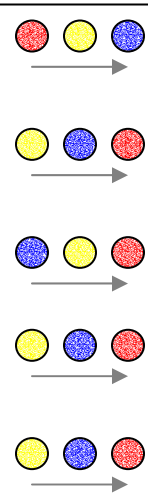
3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D	yellow		A,B	
E				

Instantiation Order :



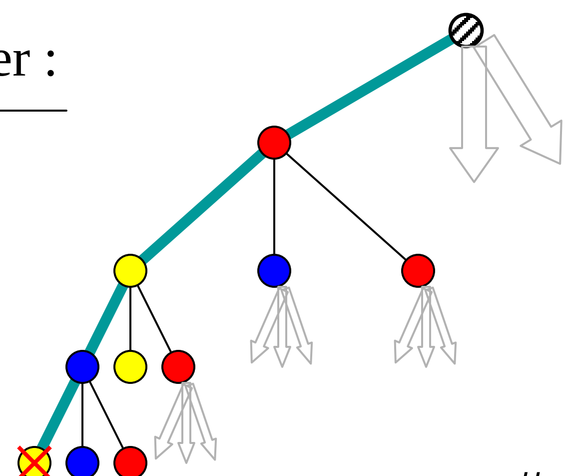
1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**

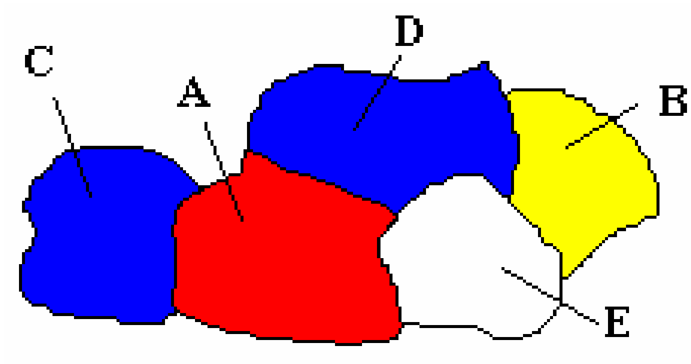


Notes:

# of Nodes Expanded

4

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	
E				

Instantiation Order :

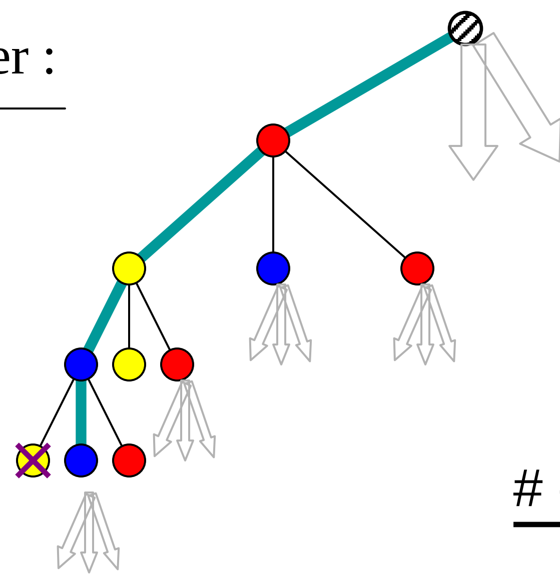
1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**

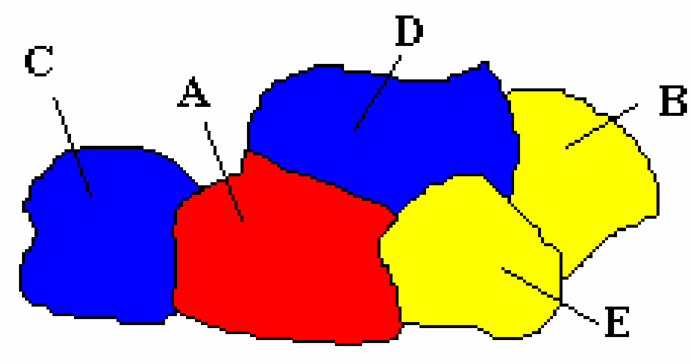


Notes:

# of Nodes Expanded

5

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	
E	yellow		A,B,D	

Notes:

Instantiation Order :

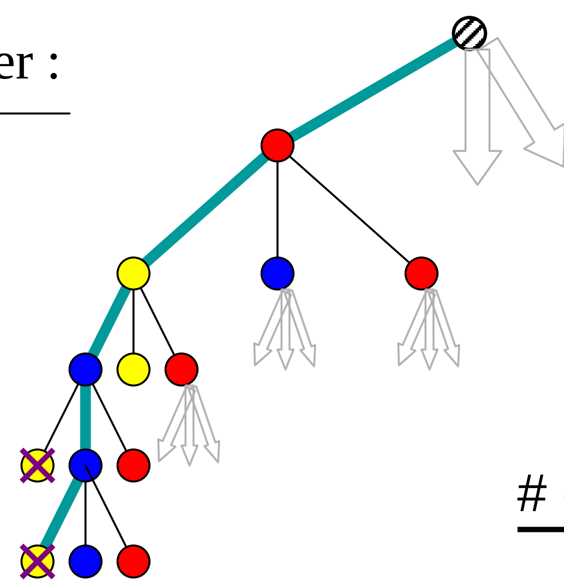
1.) A

2.) B

3.) C

4.) D

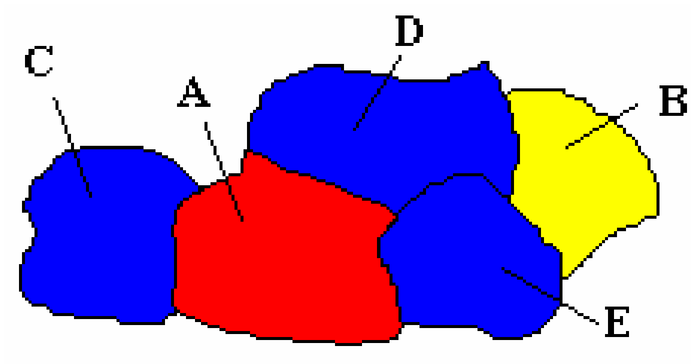
5.) E



\# of Nodes Expanded

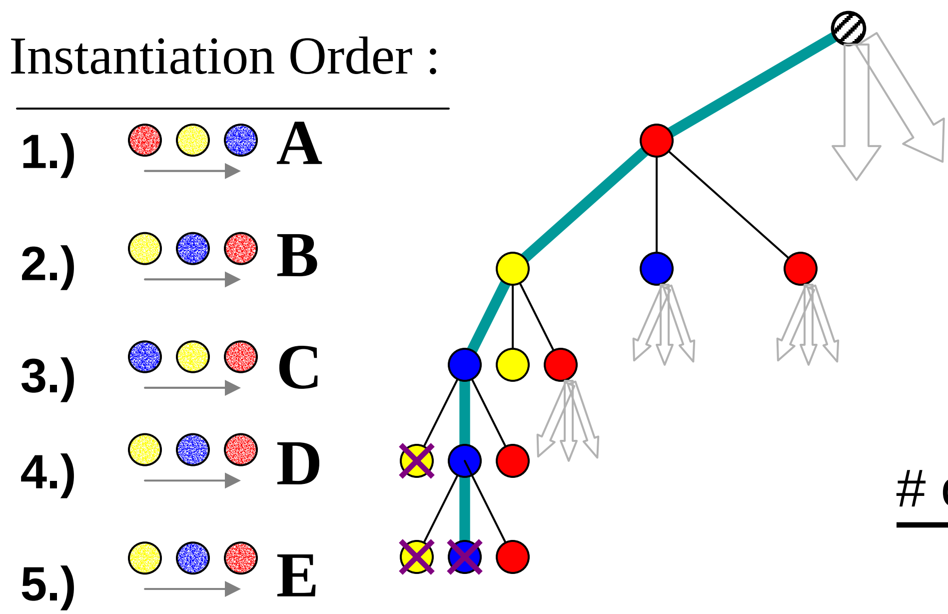
6

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	
E	yellow		A,B,D	A,B,D

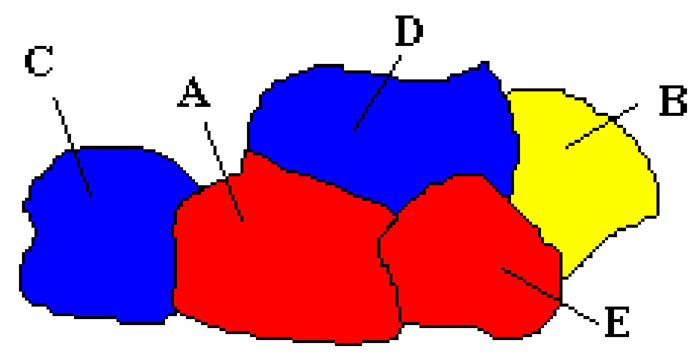


Notes:

# of Nodes Expanded

7

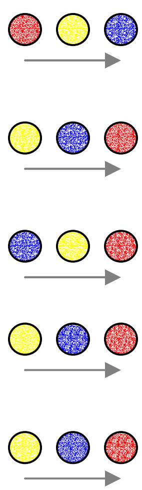
3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	
E	red	A,B,D	A,B,D	A,B,D

Instantiation Order :



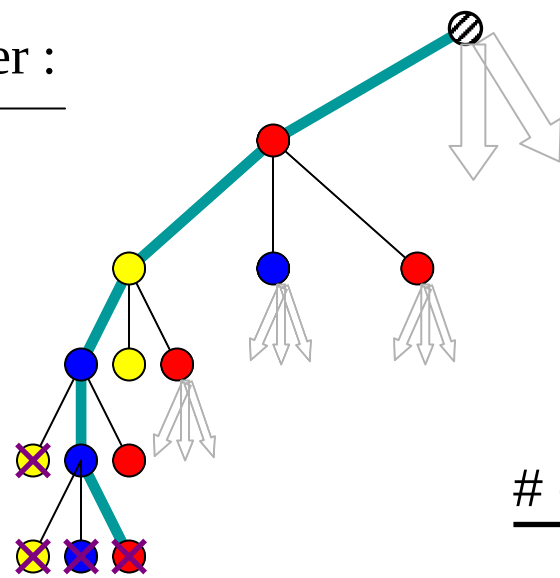
1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**

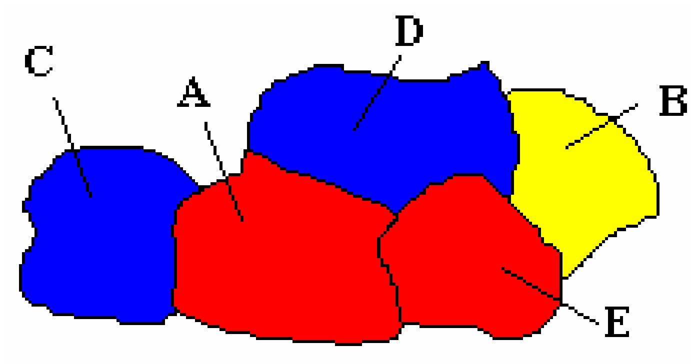


Notes:

# of Nodes Expanded

8

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	yellow			
C	blue			
D	blue		A,B	A,B
E	~~red~~	A,B,D	A,B,D	A,B,D

Instantiation Order :

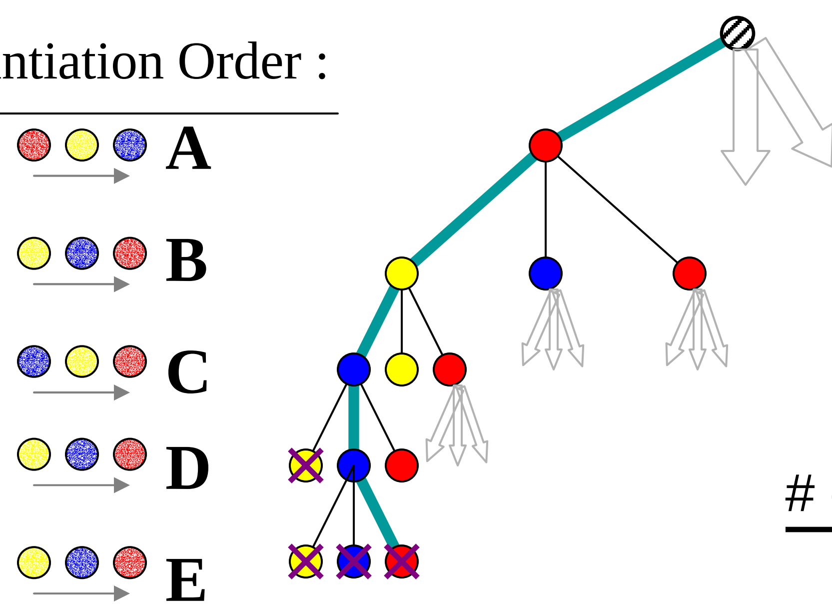
1.) A

2.) B

3.) C

4.) D

5.) E

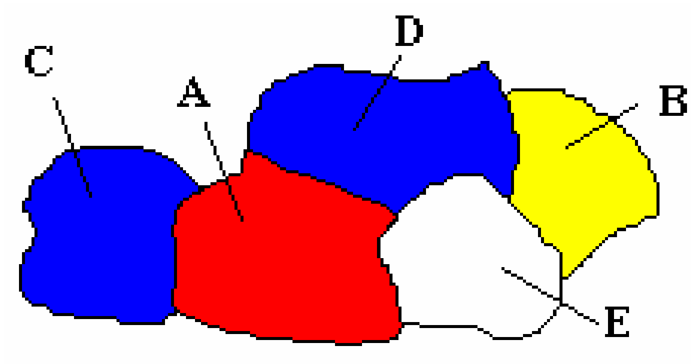


Notes:

# of Nodes Expanded

8

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	yellow			
C	blue			
D	~~blue~~		A,B	A,B
E	~~red~~			

Instantiation Order :

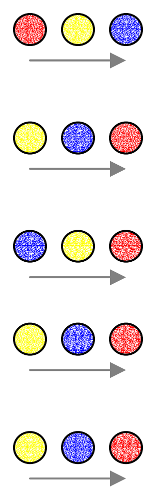
1.) **A**

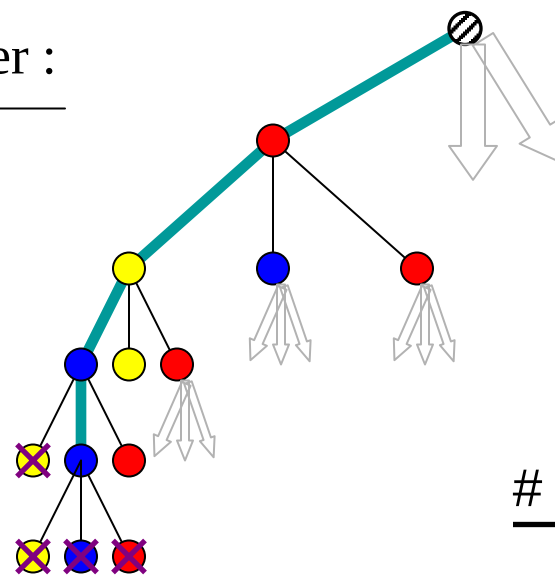
2.) **B**

3.) **C**

4.) **D**

5.) **E**



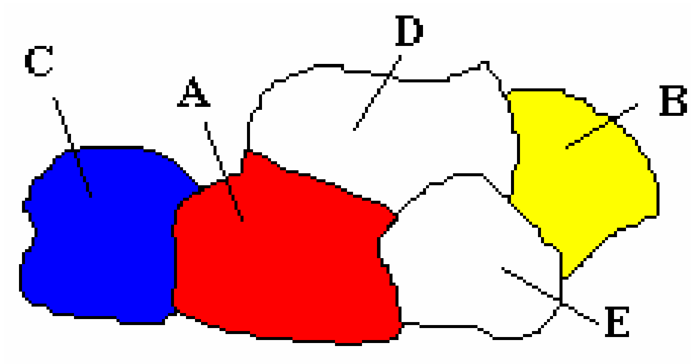


Notes:

# of Nodes Expanded

8

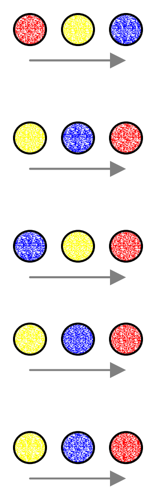
3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	yellow			
C	blue			
D	red		A,B	A,B
E				

Instantiation Order :



1.) **A**

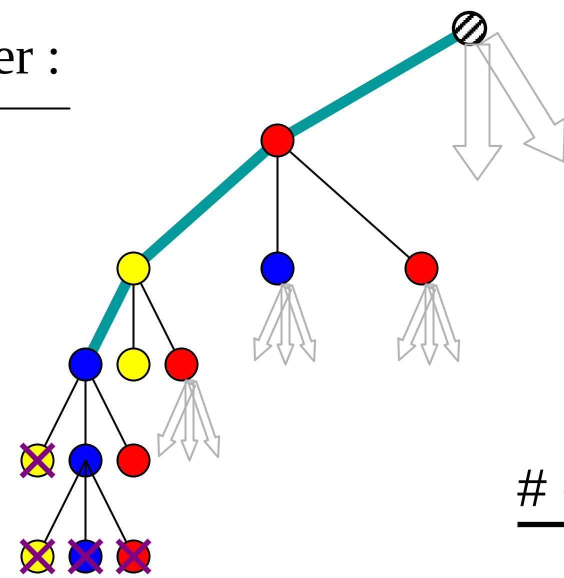
2.) **B**

3.) **C**

4.) **D**

5.) **E**

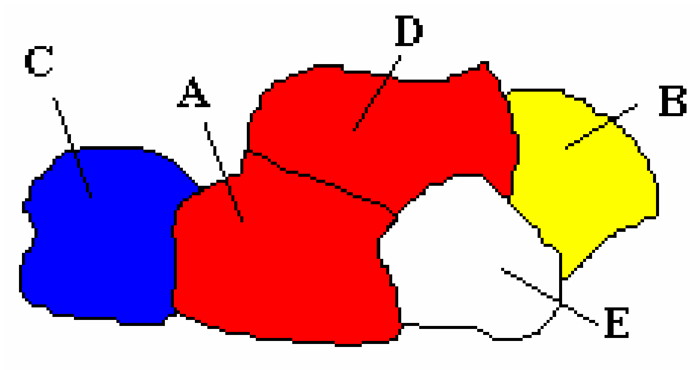
Notes:



\# of Nodes Expanded

8

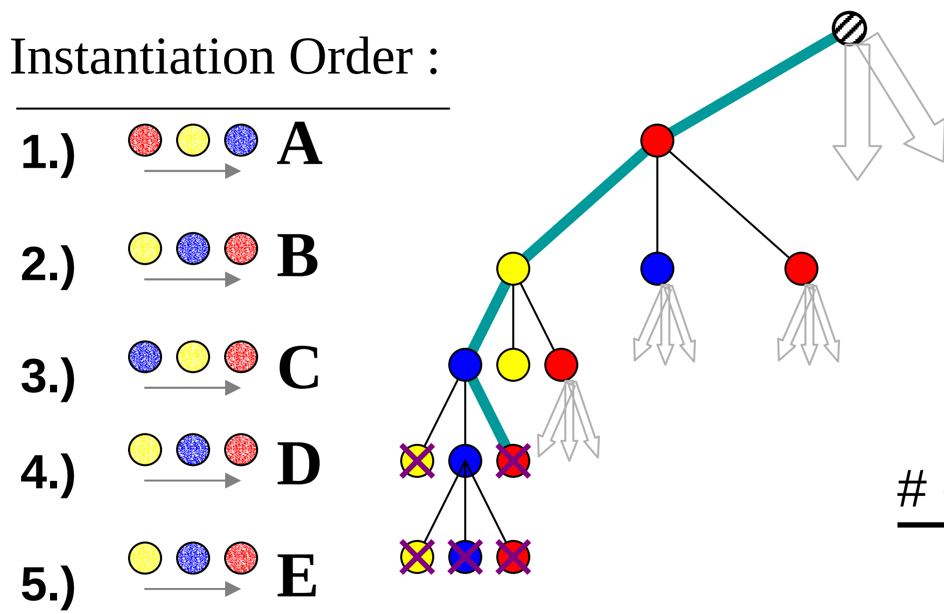
3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	~~yellow~~			
C	blue			
D	~~red~~	A,B	A,B	A,B
E				

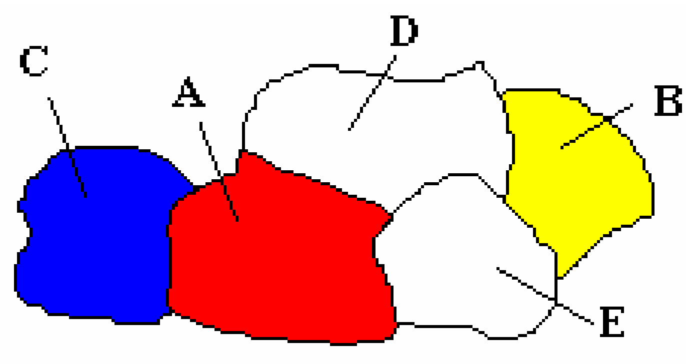
Notes:



# of Nodes Expanded

9

3.) Dynamic Backtracking



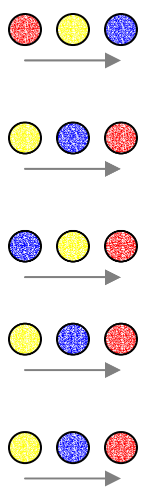
Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	~~yellow~~			
C	blue			
D		A,B	A,B	A,B
E				

Notes:

Dynamic Backtracking doesn't have to erase C.

Instantiation Order :



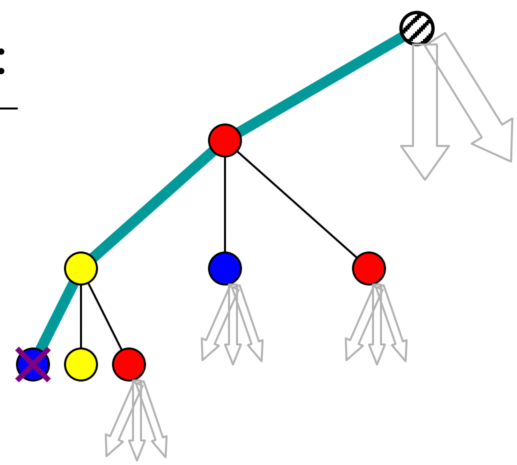
1.) A

2.) B

3.) C

4.) D

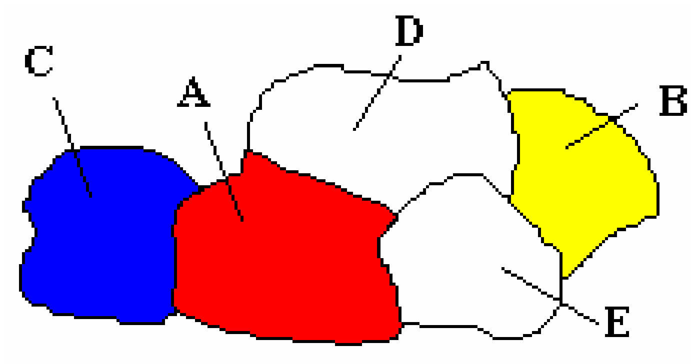
5.) E



# of Nodes Expanded

9

3.) Dynamic Backtracking



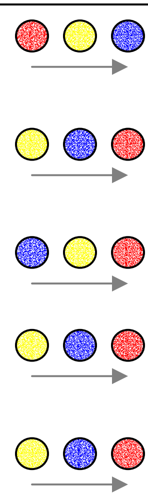
Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
B	~~yellow~~		A	
C	blue			
D		A,B	A,B	A,B
E				

Notes:

Dynamic Backtracking doesn't have to erase C.

Instantiation Order :



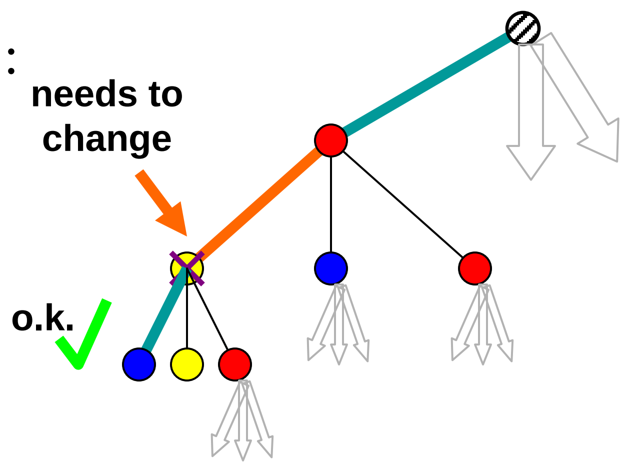
1.) A

2.) B

3.) C

4.) D

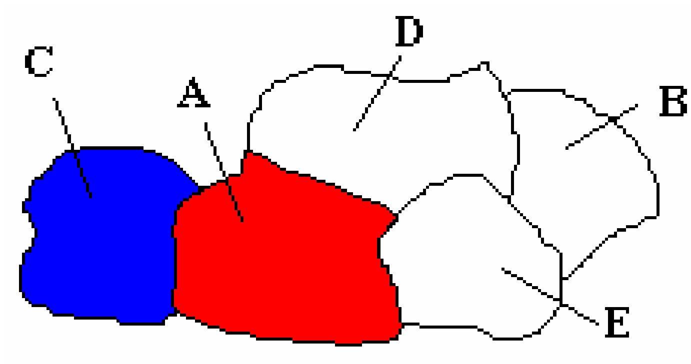
5.) E



# of Nodes Expanded

9

3.) Dynamic Backtracking



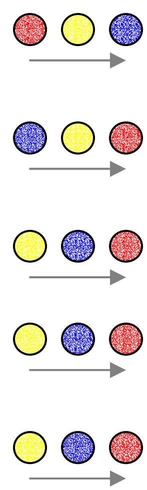
Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
C	blue			
B			A	
D				
E				

Notes:

A is recorded as the reason B can't be yellow

Instantiation Order :



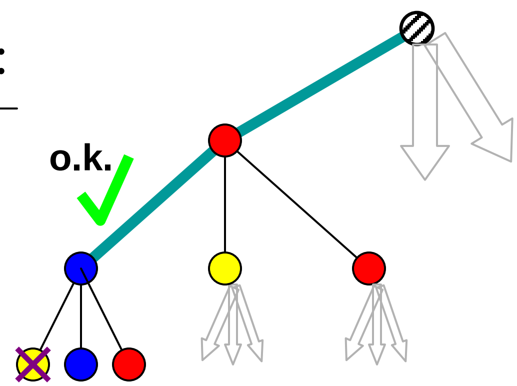
1.) A

3.) C

2.) B

4.) D

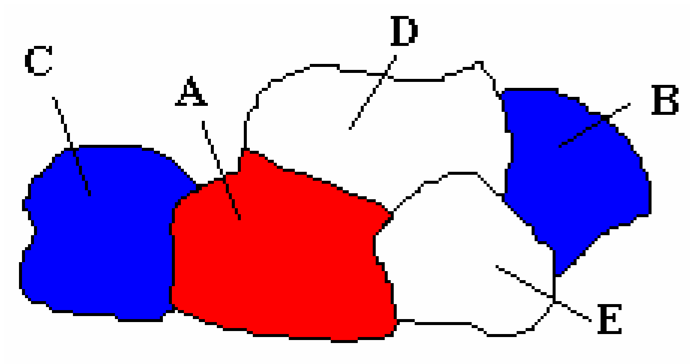
5.) E



# of Nodes Expanded

9

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
C	blue			
B	blue		A	A
D				
E				

Notes:

This is the same problem as before but only costs 6 nodes, since it is lower in the tree.

Instantiation Order :

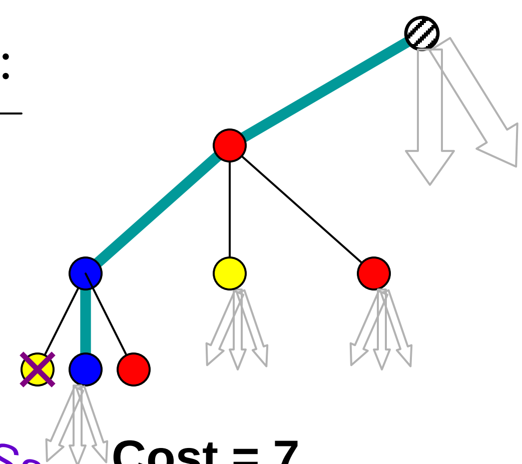
1.) A

3.) C

2.) B

4.) D

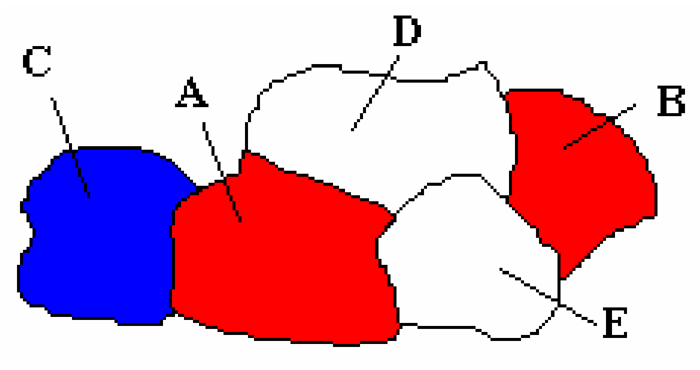
5.) E



# of Nodes Expanded

16

3.) Dynamic Backtracking



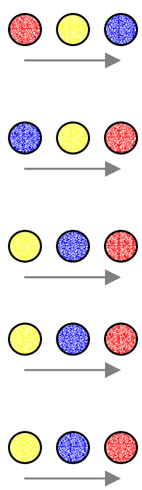
Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
C	blue			
B	blue		A	A
D				
E				

Notes:

Now we're on the right track.

Instantiation Order :



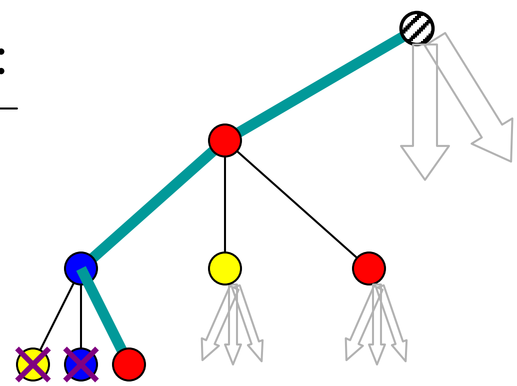
1.) A

3.) C

2.) B

4.) D

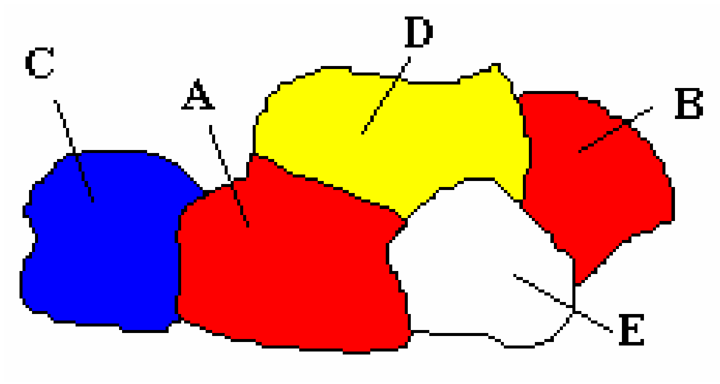
5.) E



# of Nodes Expanded

17

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
C	blue			
B	blue		A	A
D	yellow			
E				

Notes:

Now we're on the right track.

Instantiation Order :

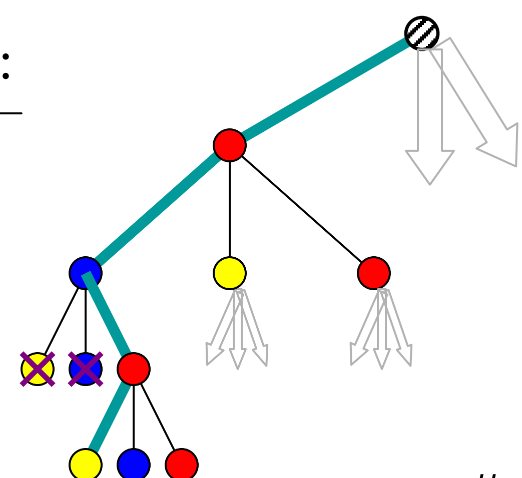
1.) A

3.) C

2.) B

4.) D

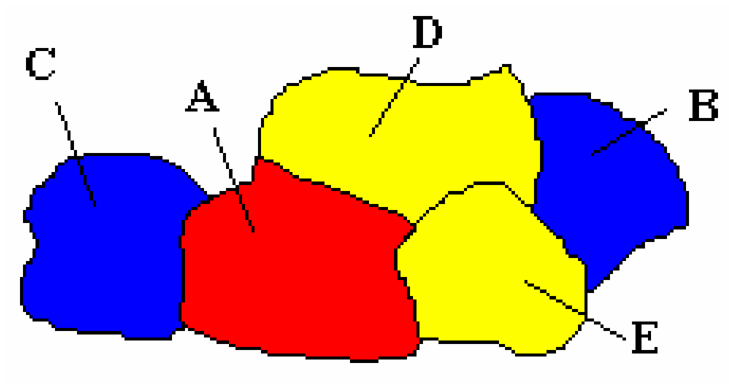
5.) E



# of Nodes Expanded

18

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
C	blue			
B	blue		A	A
D	yellow			
E			A,B,D	

Notes:

Now we're on the right track.

Instantiation Order :

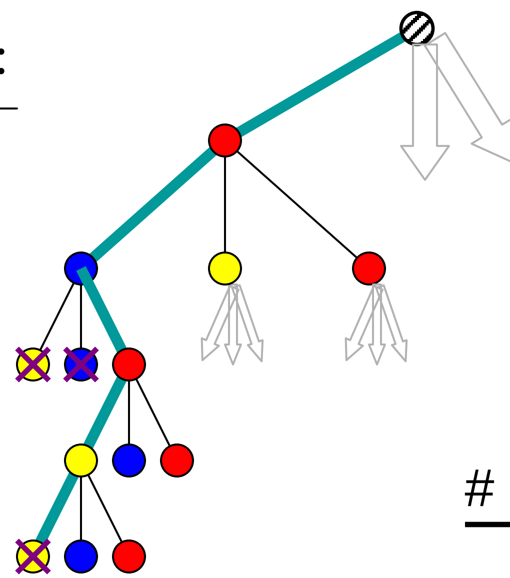
1.) A

3.) C

2.) B

4.) D

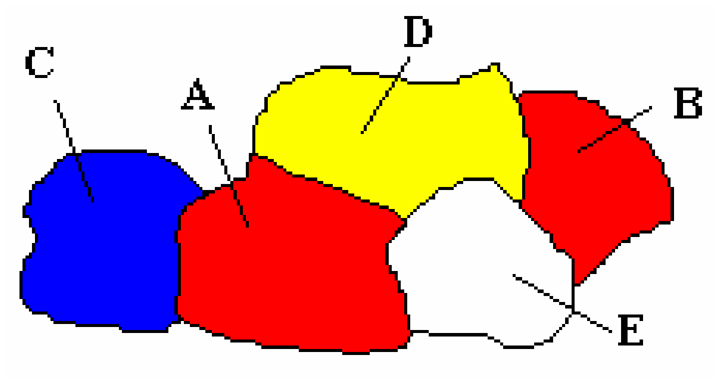
5.) E



# of Nodes Expanded

19

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
C	blue			
B	blue		A	A
D	yellow			
E			A,B,D	

Notes:

Now we're on the right track.

Instantiation Order :

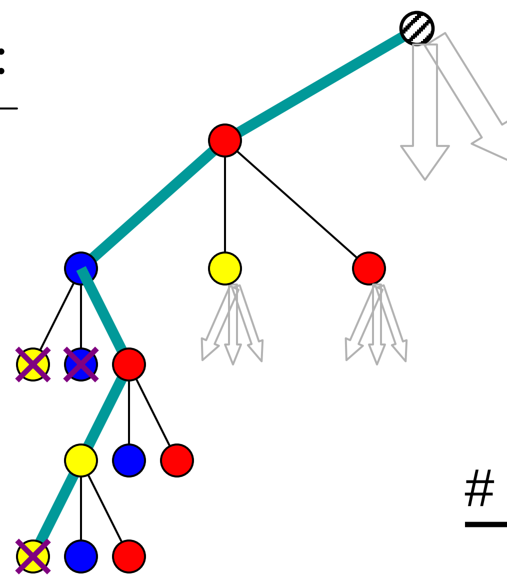
1.) A

3.) C

2.) B

4.) D

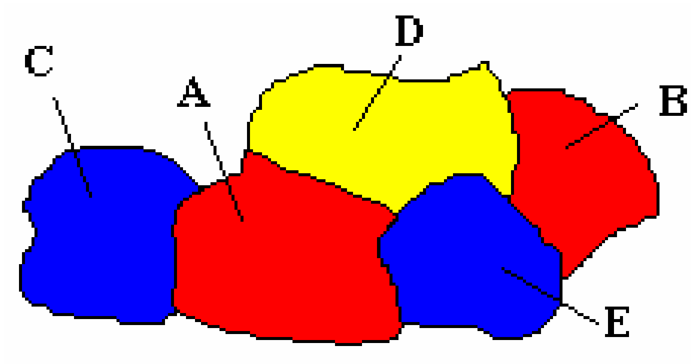
5.) E



# of Nodes Expanded

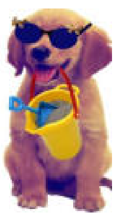
19

3.) Dynamic Backtracking



Elimination Explanations:

Region	Color	Red	yellow	blue
A	red			
C	blue			
B	blue		A	A
D	yellow			
E			A,B,D	



Notes:

Solution Found!!

Instantiation Order :

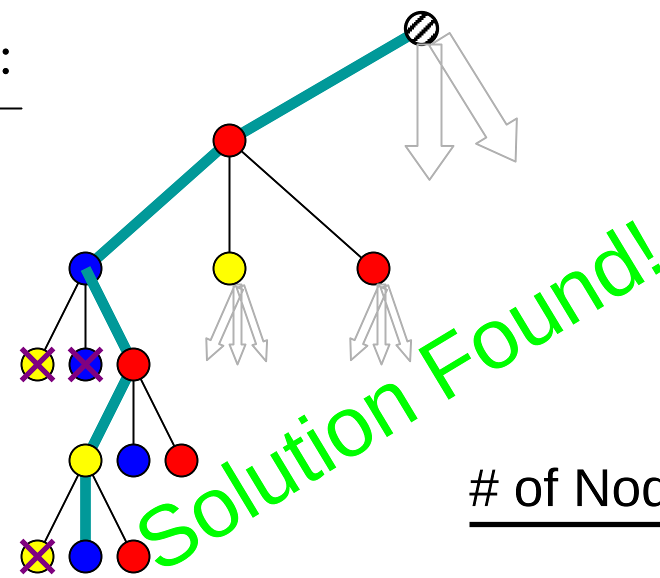
1.) A

3.) C

2.) B

4.) D

5.) E



# of Nodes Expanded

20

Dynamic Backtracking Summary

- **Positive Features**
  - Allows a Variable Instantiation Order
  - Allows partial assignments to remain assigned (if they are not part of a conflict)

- **Negative Features**
  - Requires a conflict-detection sub-routine
  - Requires more memory than simple backtrack search
  - Completeness proof is not easy to understand or visualize (the proof that it doesn't skip over any of the search space)

Map Example Results	
Chronological BT	37
Conflict-Directed BT	23
Dynamic BT	20

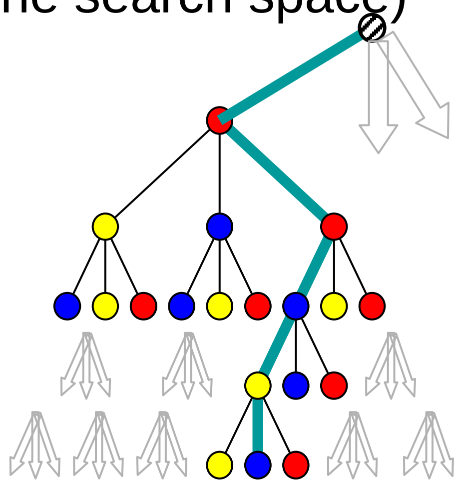
1.) **A**

2.) **B**

3.) **C**

4.) **D**

5.) **E**



Any Questions?