



配属の数理(1)

良いペアを作ろう！

ここで学ぶこと

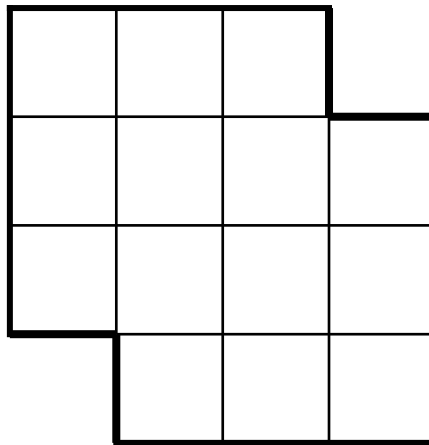
- マッチング問題
 - 市松模様で色分け可能(2部グラフ)の場合
 - 一般グラフの場合
- 割当問題 誰にどの仕事を割り当てる?
- 配属問題 ゼミの配属を決めよう
- 安定結婚問題 不満を抑えるマッチング

例題1

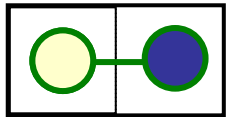
畳の敷き詰めプランを作成しよう



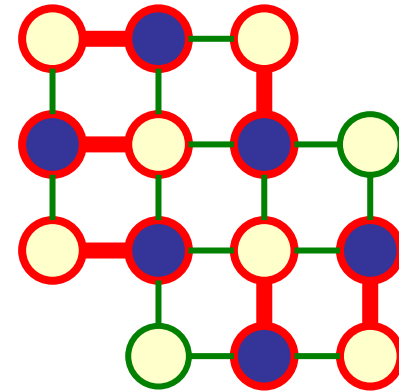
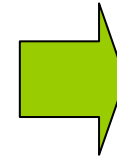
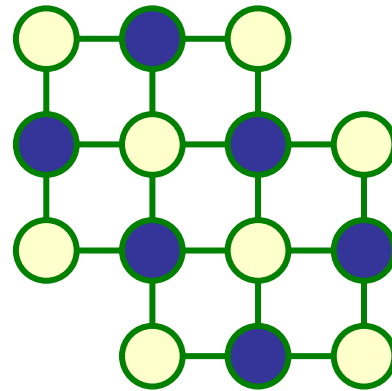
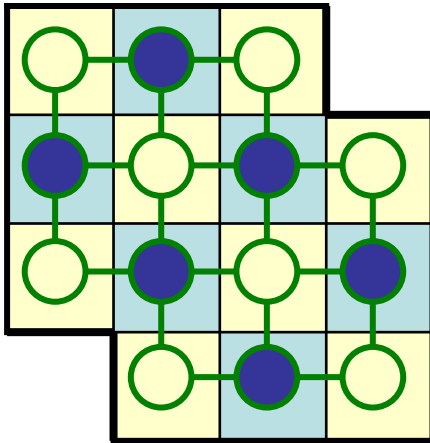
畳1枚



例題1 解説



市松模様塗れる
↓
マッチングを見つけやすいぞ!



● 6点 < ○ 8点

マッチング数6

市松模様

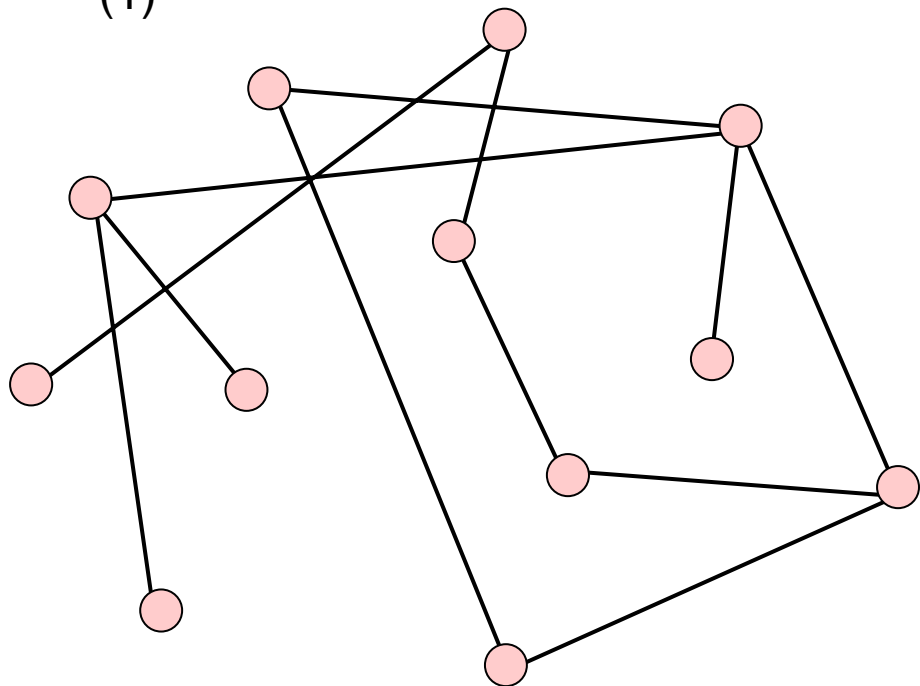
高々6枚しか畳は置けない

最大マッチング

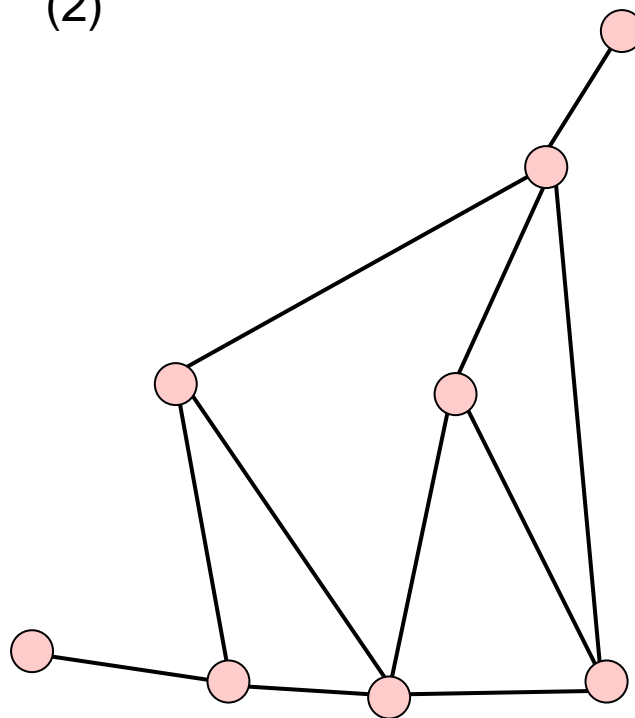
市松模様(Shimoda)に塗ってみよう!



(1)

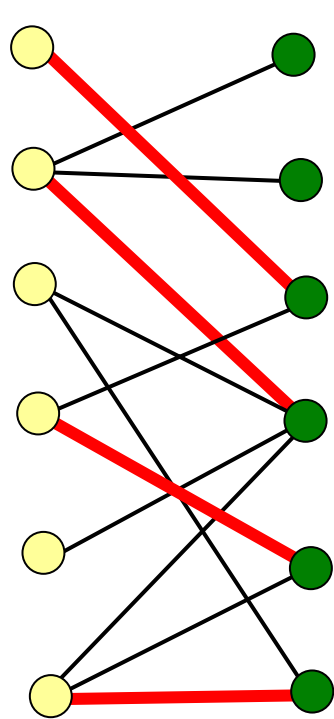


(2)

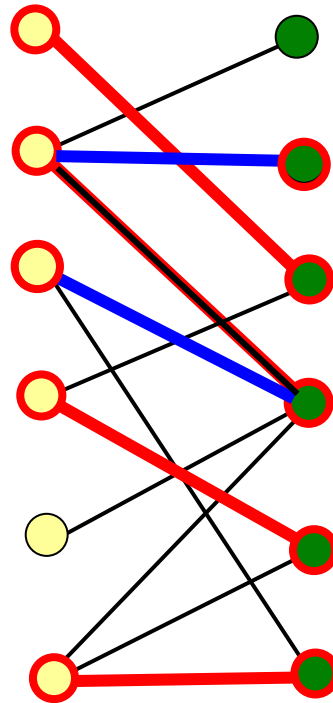


2部グラフ上の最大マッチング

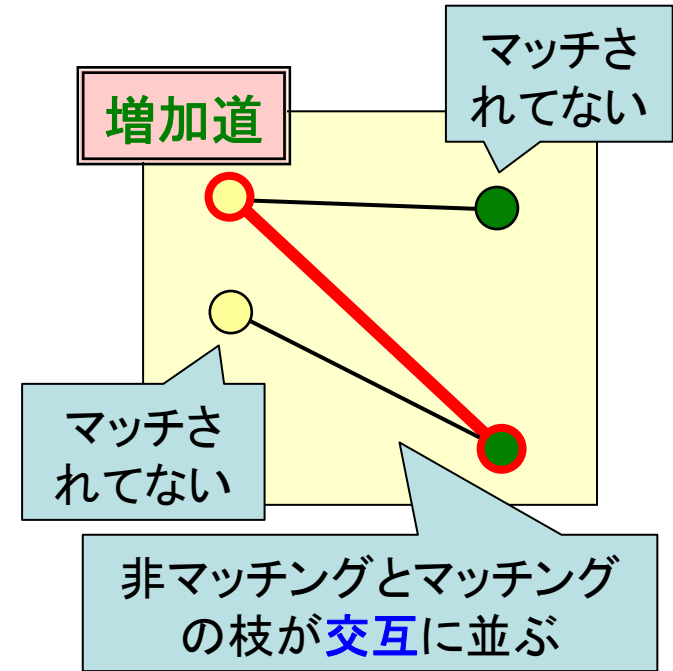
マッチングの例



注目
マッチされて
いない点



マッチング数を増やせる場合



Q. 最大マッチング?

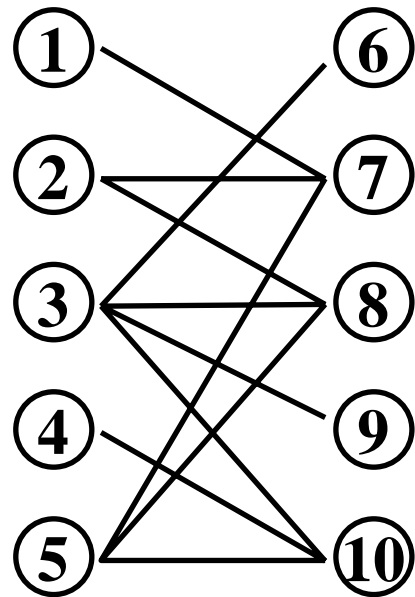


増加道を見つけて、マッチングを増やす
増加道が無ければ、最大マッチング

→ 増加道法

練習 Shall we dance?

ダンスパーティーに男性・女性各5人が集まった。
パートナーになりたい希望は以下の組合せである。



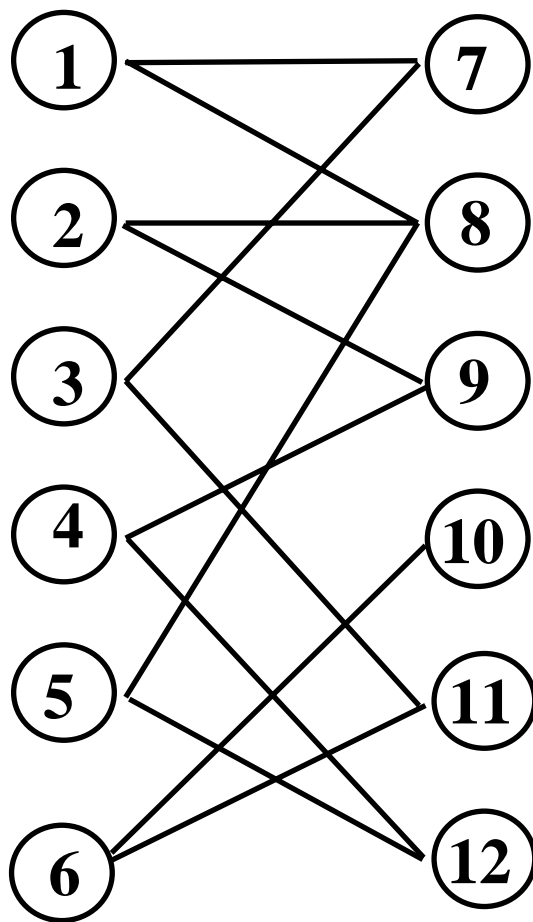
男性

女性

さて、なるべく多くのペアを組みたいが
最大で何組できるか？その組み方は？



演習1 最大マッチングを求めよう



演習2 バス会社運行係

- 6ルート of バス運行を計画中
- 各ターミナル間の回送時間は右下表の通り
- 最低何台で運行可能？

	発地	出発時間	着地	到着時間
ルート①	A	9:20	C	9:40
ルート②	B	10:00	A	10:30
ルート③	B	8:40	B	9:50
ルート④	D	8:00	B	8:30
ルート⑤	C	12:30	E	13:30
ルート⑥	E	11:10	C	12:20

計画バスルート

回送着地

	A	B	C	D	E
A	0	3	1	6	2
B	4	0	5	5	6
C	2	5	0	6	4
D	3	2	1	0	5
E	7	3	5	4	0

回送発地

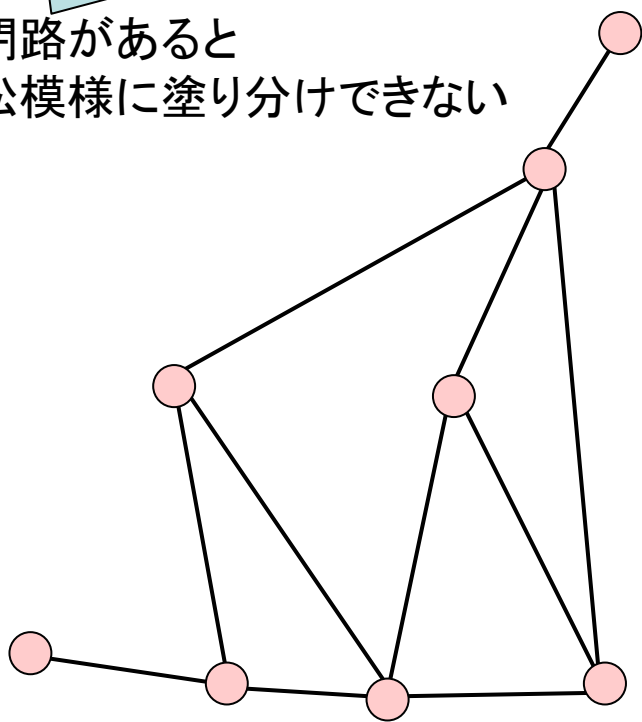


(単位:10分)

発展 2部グラフでない場合

2部グラフではない

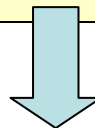
奇閉路があると
市松模様に塗り分けできない



困った, 困った

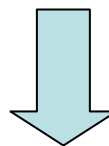
1965年

奇閉路を見つけたら
一時的に一点にみなせば
何とかなるんじゃない?



花アルゴリズム

最小重み最大マッチングを
見つけることも可能



詳しくは
もっと勉強するのじゃ

Edmons博士



組合せ最適化問題に
大きな影響を与える

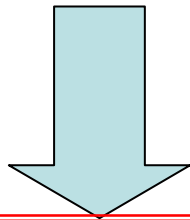
ここまでのまとめ

奇閉路が無いグラフ

- **最大マッチング**を求める
 - 市松模様で色分け可能(**2部グラフ**)の場合
 - 一般グラフの場合

花アルゴリズム

増加道法



次は枝に**重み**のある場合の話題

- **割当問題** 誰にどの仕事を割り当てる?
- **配属問題** ゼミの配属を決めよう
- **安定結婚問題** 不満を抑えるマッチング

例題2 仕事の割当

5つの支社へ一人ずつの人員補強を計画。
希望任地と、その任地への赴任費用は下表のとおり。

	支社①	支社②	支社③	支社④	支社⑤
Aさん	25	30			
Bさん	20		70	35	
Cさん	80	75	90	65	
Dさん				55	40
Eさん				60	50

空白は希望
しない支社

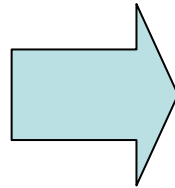
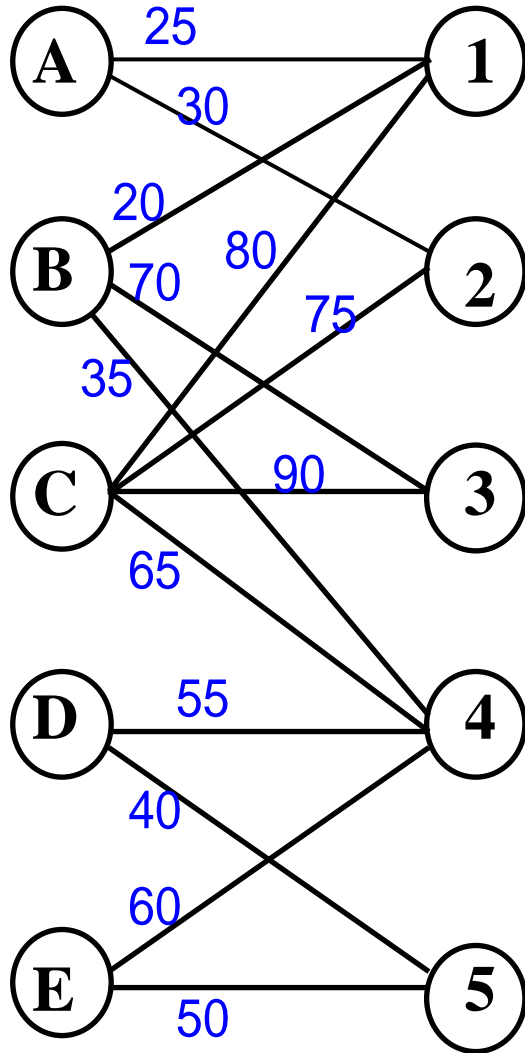


さて、誰をどの支社に配属すれば最も費用が安く済む？

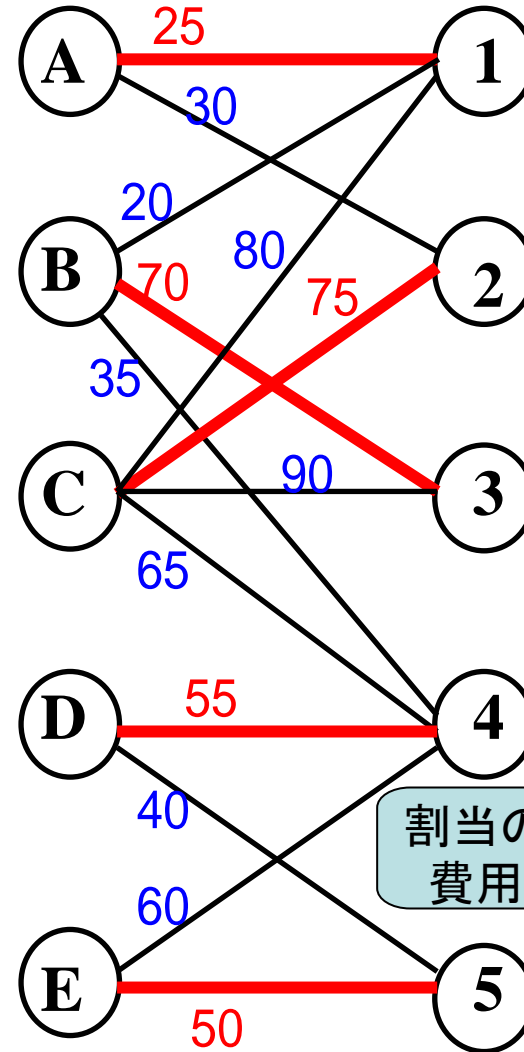
関連問題：5人を各支社に割り当てることはできるか？

問題の図示

割当の総費用



適当に割当



25
+
70
+
75
+
55
+
50
||

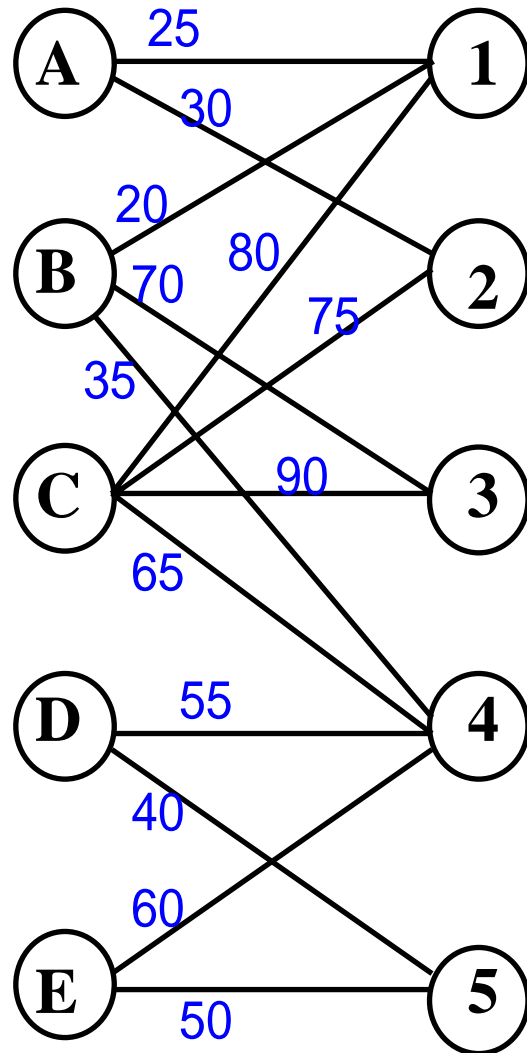
275

割当の
費用

割当問題

費用が最小になる割当は?

練習 最適割当を求めよう



最適割当を求める方法

解法はいくつも提案

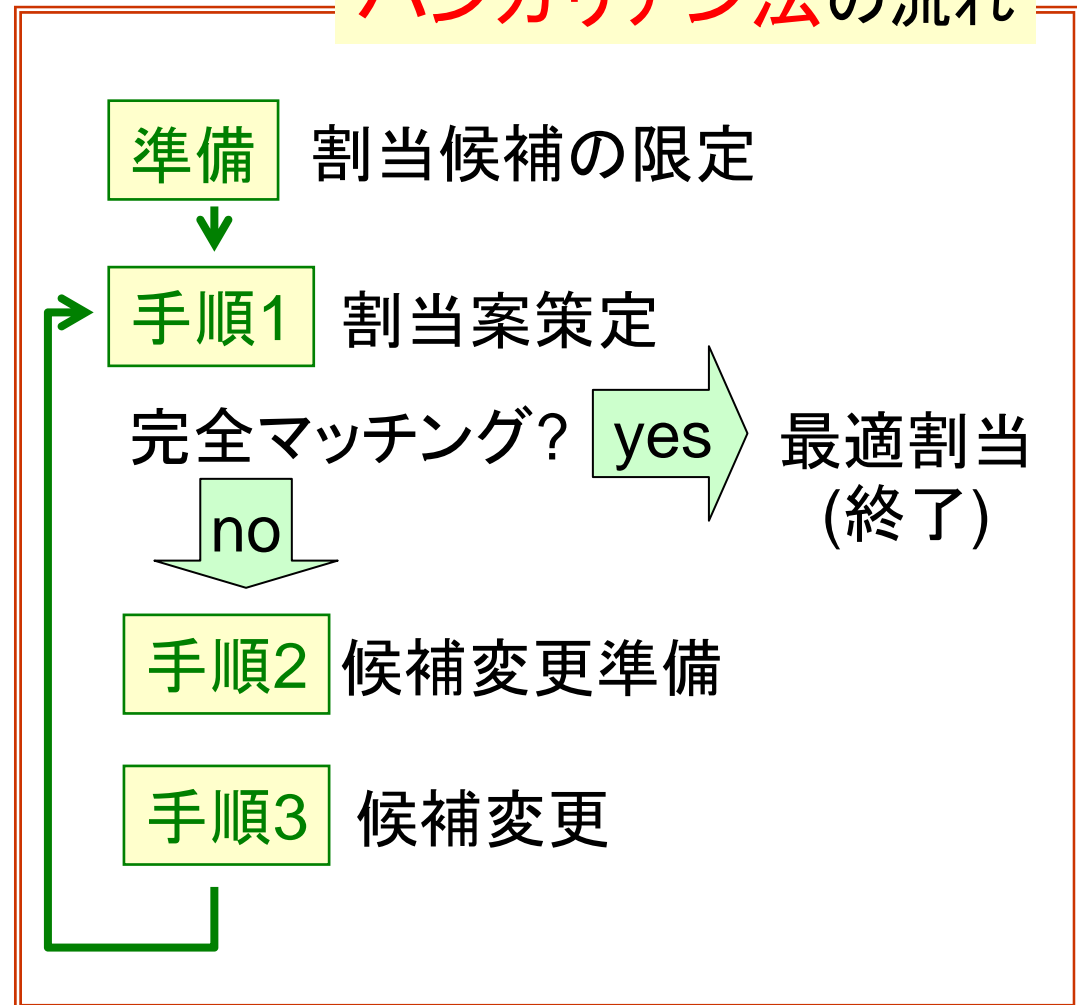
紹介

- オークション法
- ハンガリアン法
- 最小費用流問題

必要なもの: 表

	①	②	③	④	⑤
A	25	30			
B	20		70	35	
C	80	75	90	65	
D				55	40
E				60	50

ハンガリアン法の流れ



準備 割当候補の限定

①行毎に最小数字を発見

	①	②	③	④	⑤
A	25	30			
B	20		70	35	
C	80	75	90	65	
D				55	40
E				60	50

②行毎に最小数字を引く

	①	②	③	④	⑤
A	0	5			
B	0		50	15	
C	15	10	25	0	
D				15	0
E				10	0

③列毎に最小数字を発見

	①	②	③	④	⑤
A	0	5			
B	0		50	15	
C	15	10	25	0	
D				15	0
E				10	0

④列毎に最小数字を引く

	①	②	③	④	⑤
A	0	0			
B	0		25	15	
C	15	5	0	0	
D				15	0
E				10	0

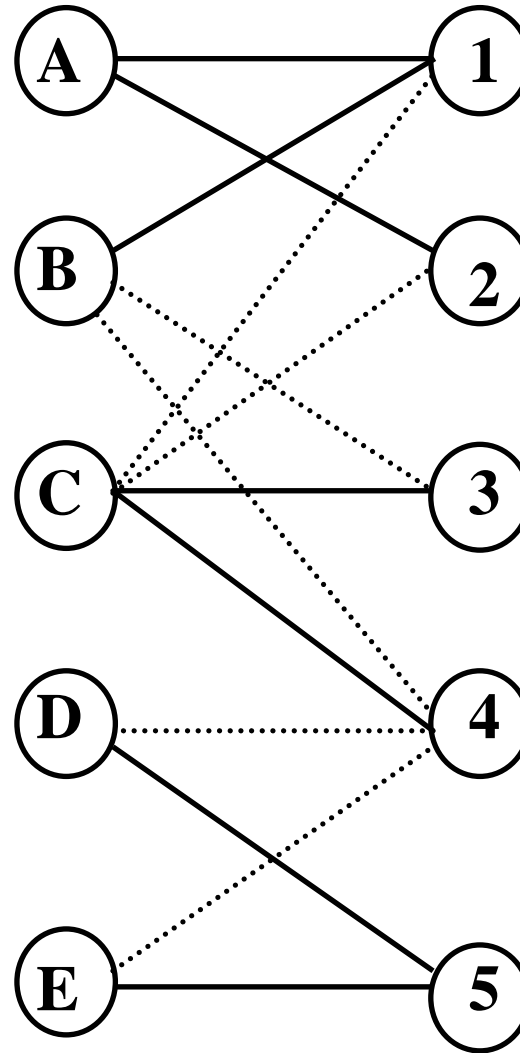
準備完了

手順1 割当案の作成

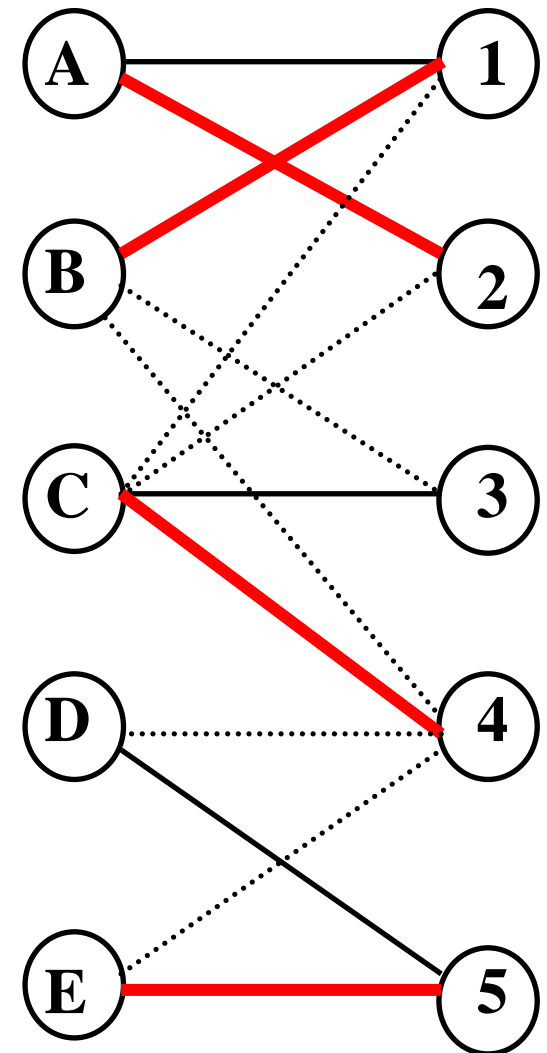
「0」部分だけで
最大マッチングを求める

	①	②	③	④	⑤
A	0	0			
B	0		25	15	
C	15	5	0	0	
D				15	0
E				10	0

	①	②	③	④	⑤
A	0	0			
B	0		25	15	
C	15	5	0	0	
D				15	0
E				10	0



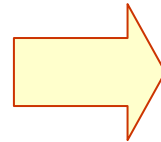
「0」部分のみ抽出



最大マッチング

手順2 割当候補の変更準備

	①	②	③	④	⑤
A	0	0			
B	0		25	15	
C	15	5	0	0	
D				15	0
E				10	0



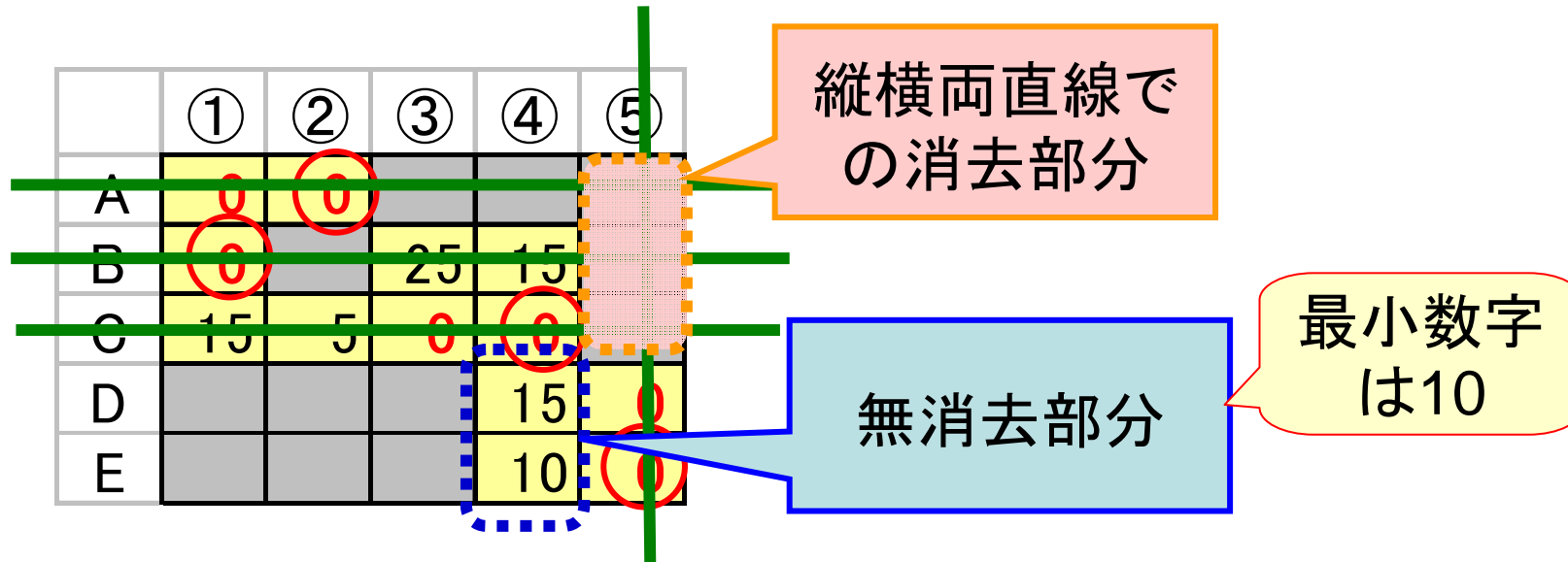
	①	②	③	④	⑤
A	0	0			
B	0		25	15	
C	15	5	0	0	
D				15	0
E				10	0

準備完了

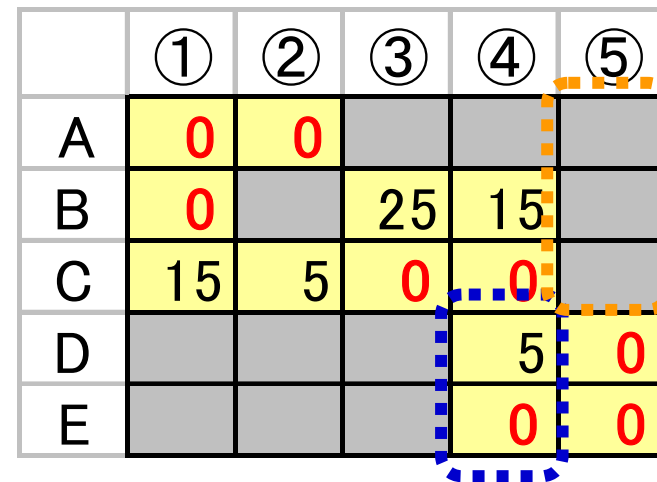
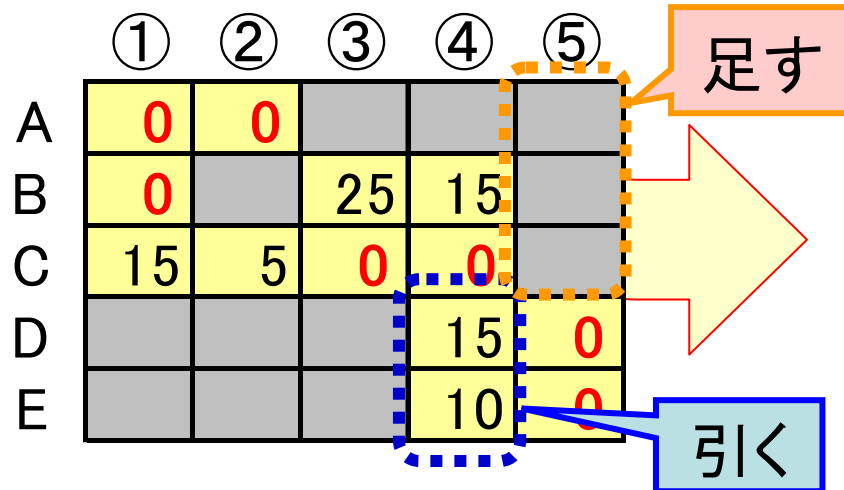
○部分から縦か横に直線を引き、
すべての「0」を線で消す

※線の引き方は複数通り存在
どれでも良い

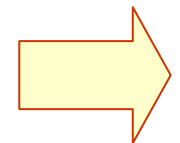
手順3 割当候補の変更



無消去部分の最小数字を



手順1へ戻る

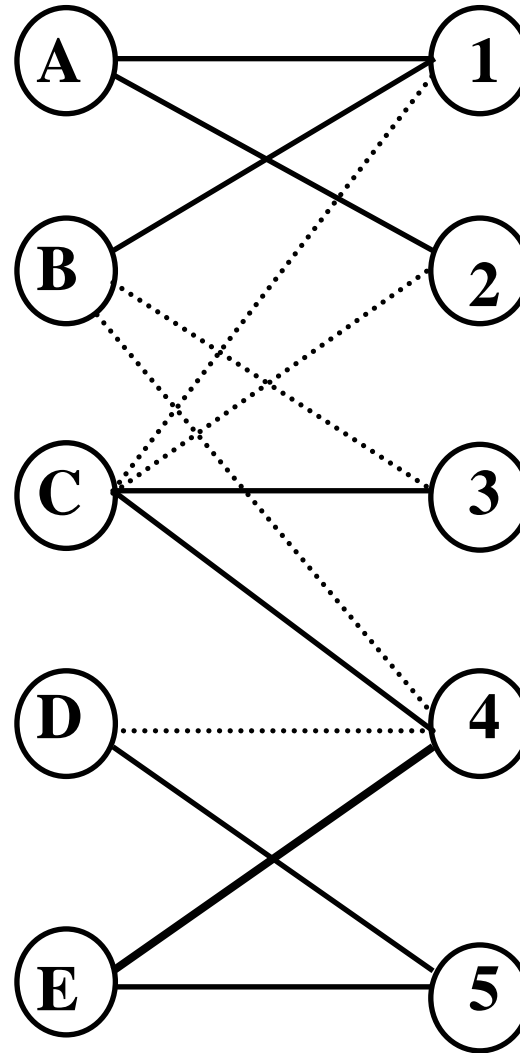


手順1 割当案の作成

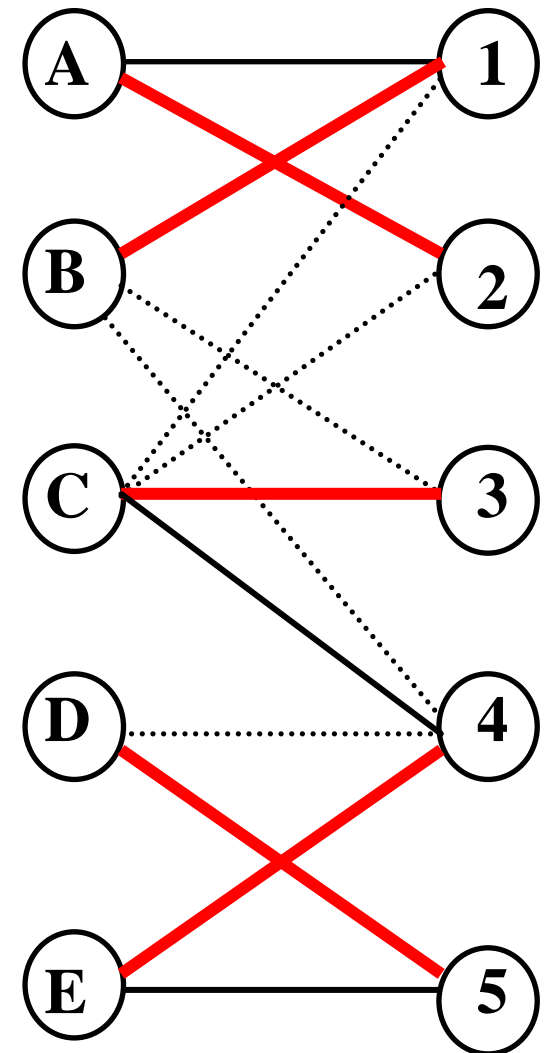
「0」部分だけで
最大マッチングを求める

	①	②	③	④	⑤
A	0	0			
B	0		25	15	
C	15	5	0	0	
D				5	0
E				0	0

	①	②	③	④	⑤
A	0	0			
B	0		25	15	
C	15	5	0	0	
D				5	0
E				0	0



「0」部分のみ抽出



完全マッチング

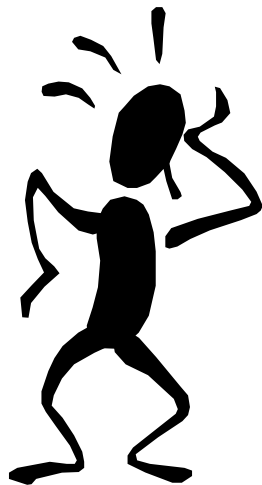
最適割当の導出

得られた割当

	①	②	③	④	⑤
A	0	0			
B	0		25	15	
C	15	5	0	0	
D				5	0
E				0	0

総コストが最小な割当
(最適割当)

	①	②	③	④	⑤
A	25	30			
B	20		70	35	
C	80	75	90	65	
D				55	40
E				60	50



どうして発見で
きたのだろう？

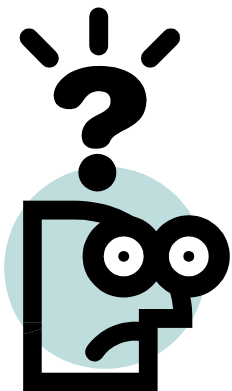
$$20+30+90+60+40=240$$

演習3

資格の必要な4つの仕事のために4人採用した。
保有資格のランクが異なり、仕事により給料が変わる。
人件費を最小にするには誰にどの仕事を割当ててる？

	仕事①	仕事②	仕事③	仕事④
Aさん	45	無資格	無資格	30
Bさん	50	55	15	無資格
Cさん	無資格	60	25	75
Dさん	45	無資格	無資格	35

(単位:万円)



演習4

ある課の課長は、5人の部下A～Eと5つの異なる仕事を持って
いるが、これらの仕事は、その仕事を行なう部下との組合せで
必要とする時間が異なってくる。今、5つの仕事をP～Tとしたと
き、A～Eが各仕事に必要とする時間数は表のとおりである。部
下1人に1つの仕事を割り当て、全体で要する時間を最小にす
る時、時間の合計はいくらか。(国家I種, 平成9年度改題)

仕事に必要とする時間					
	P	Q	R	S	T
A	5	5	8	5	5
B	4	5	9	7	11
C	4	4	6	6	11
D	4	3	11	8	11
E	2	3	4	6	9



ここまでのまとめ

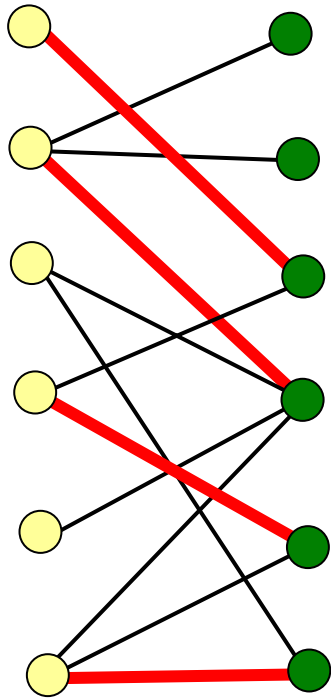
- 最大マッチングを求める
 - 市松模様で色分け可能(2部グラフ)の場合
 - 一般グラフの場合

ハンガリアン法

- 割当問題 誰にどの仕事を割り当てる?
- 配属問題 ゼミの配属を決めよう
- 安定結婚問題 不満を抑えるマッチング

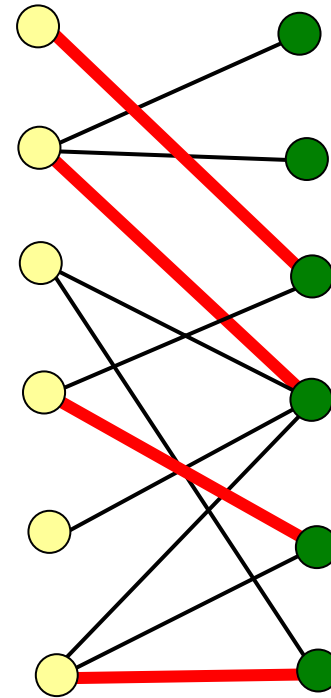
寄り道 増加道の見つけ方

奥を優先して探索



→ 奥優先探索法
Depth first search

幅を優先して探索

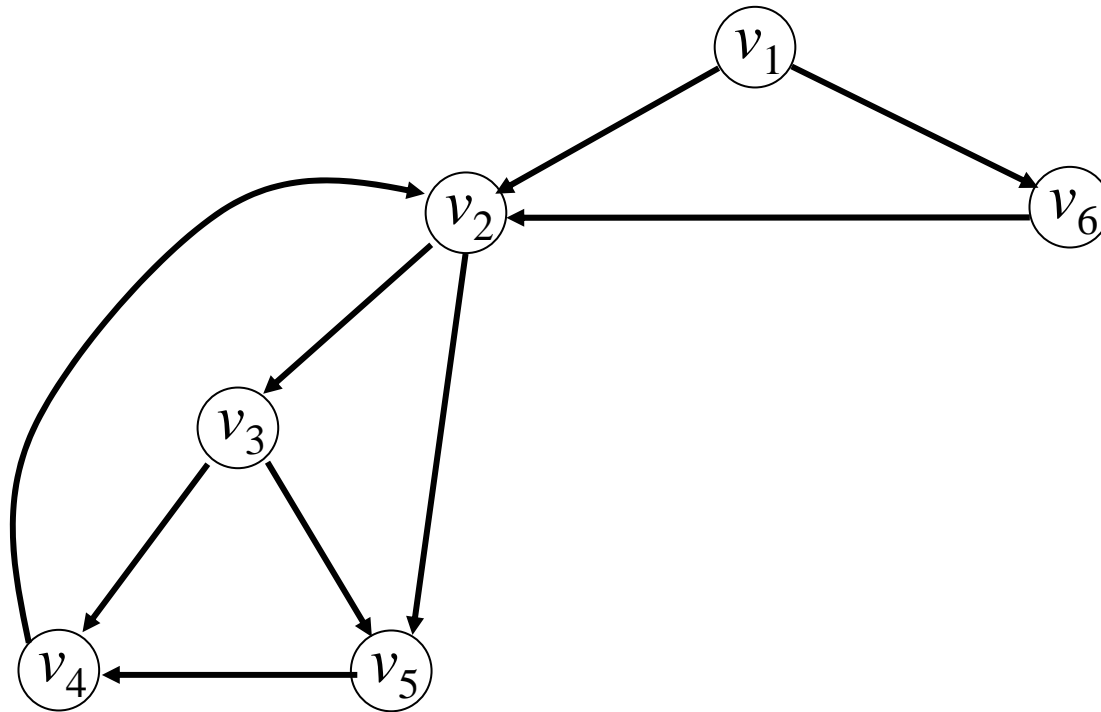


→ 幅優先探索法
Width first search

より一般的に捉えると...

グラフの探索

グラフ上のすべての点と枝を走査すること



効率の良いグラフの探索方法

奥優先探索

Depth-first search

幅優先探索

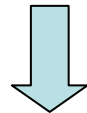
Breadth-first search

Step1: 出発点にラベル付ける

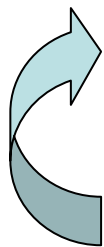
以下のStep2, 3を未探索枝がなくなるまで繰り返す。



Step2: 最新ラベルを持つ点
から未探索枝を走査



Step3: 枝の終点にラベルが無いならラベルを付け

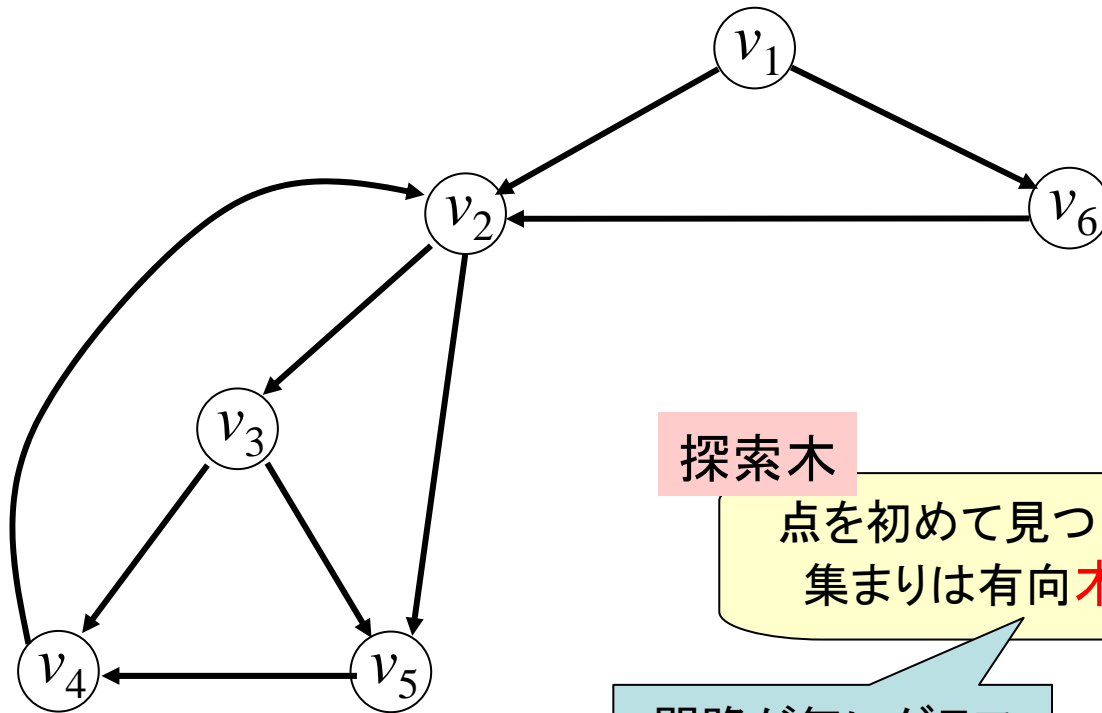


Step2: 最古ラベルを持つ点
から未探索枝を走査



例1 奥優先探索をしてみよう

出発点: v_1



探索木

点を初めて見つけた枝の
集まりは有向木になる

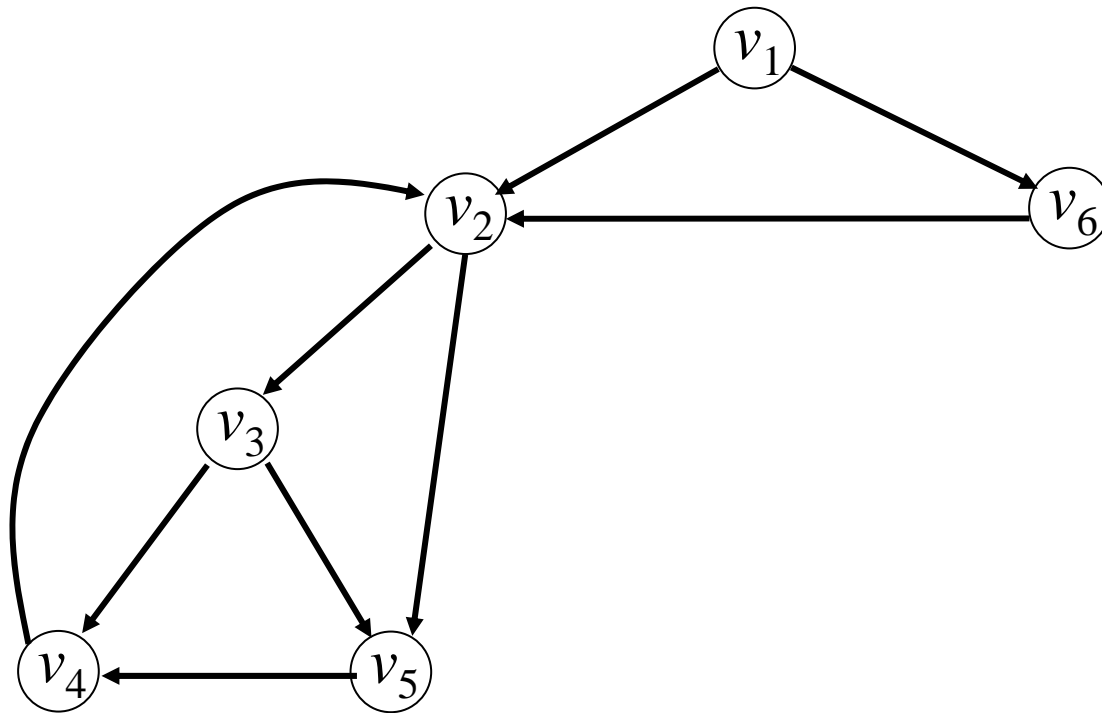
閉路が無いグラフ

探索木は有益な情報を
提供することが多い



例2 幅優先探索を試みよう

出発点: v_1



探索木は?

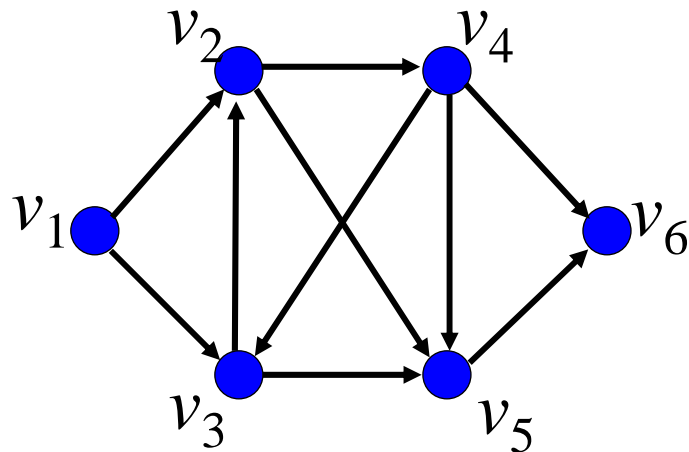
探索木利用 点への順序付け

- **前順**(先行順 : pre order)
 - ラベルと同時に番号付け
- **後順**(後行順 : post order)
 - 親の点に戻るときに番号付け



他にも
中間順 (in order)
幅優先順 (breadth-first order)
などがある

演習5 グラフの探索



出発点： v_1 として右のグラフを以下の方法で探索せよ.

- (1) 奥優先探索
- (2) 幅優先探索

また, 各々の探索木を示せ.

(3) 奥優先探索での探索木を利用し, 各点に前順・後順を各々付けてみよう。