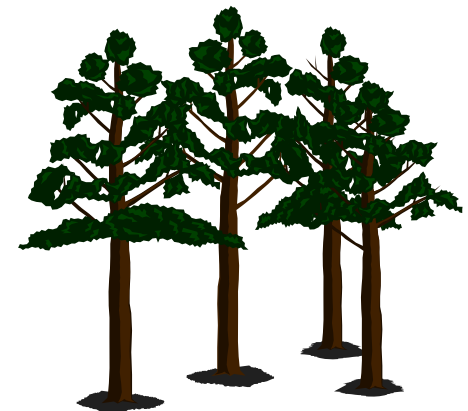




Network programming I

Minimum spanning tree problem

最適に繋げる方法



ネットワーク上に生じる様々な問題

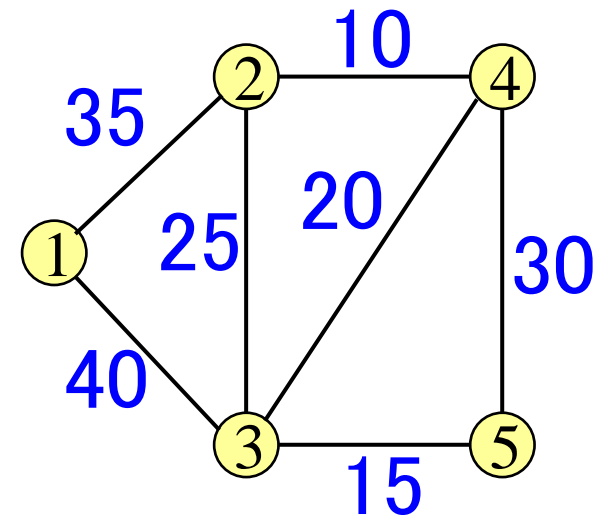
- 安価なネットワークの構築 (**最小木問題**)
- 最短ルートの探索 (**最短路問題**)
- 物の効率的な流し方 (**フロー問題**)
 - なるべく多く流す (**最大流問題**)
 - 格安に流す (**最小費用流問題**)
- どこに倉庫を配置するか (**施設配置問題**)

⇒ 事例は数限りない

→ **システム的アプローチ**が有効！

例題4-1 文教町のガス管配置

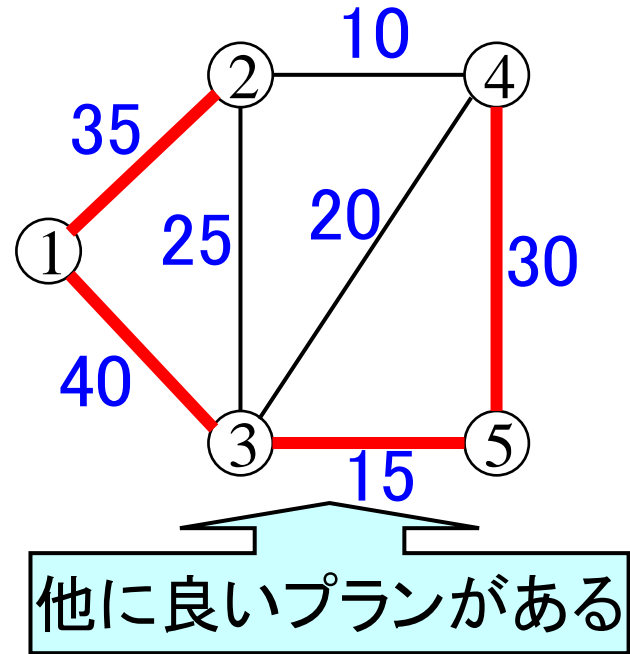
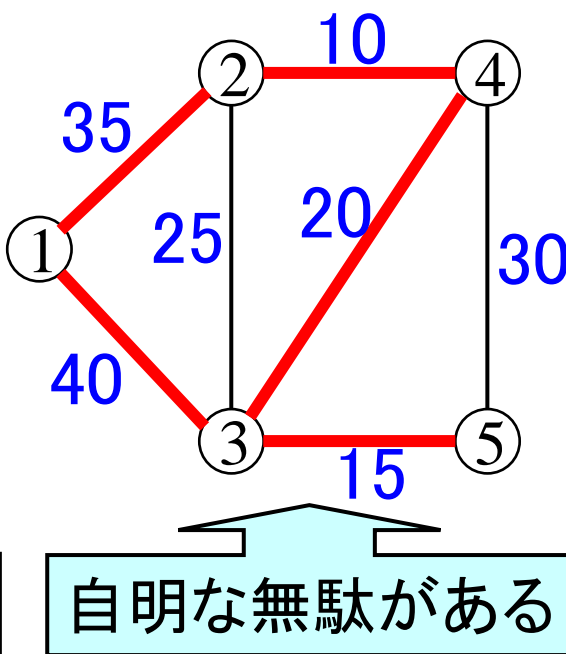
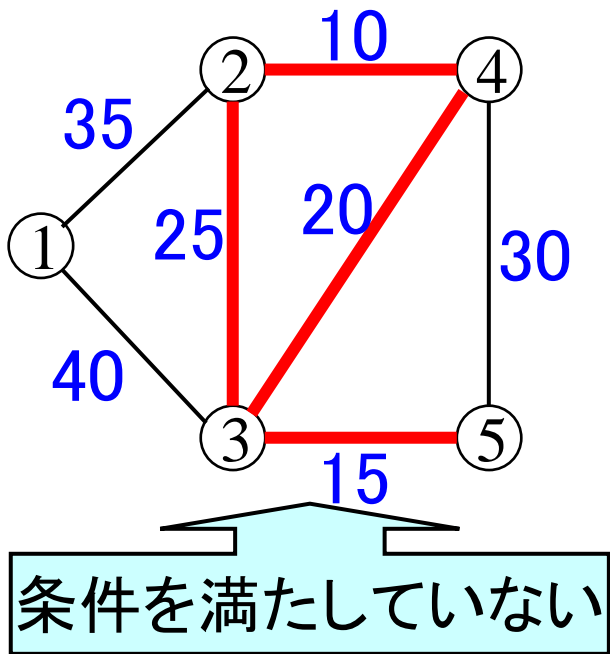
5件の家にガス管を引く。
どのようにガス管を設置
すれば費用最小??



枝: 設置可能路線
数字: 設置費用

経済的でない例

なぜ最小費用でないのか？



実行不能

改善策

閉路は無駄
× 閉路上の最大重み枝

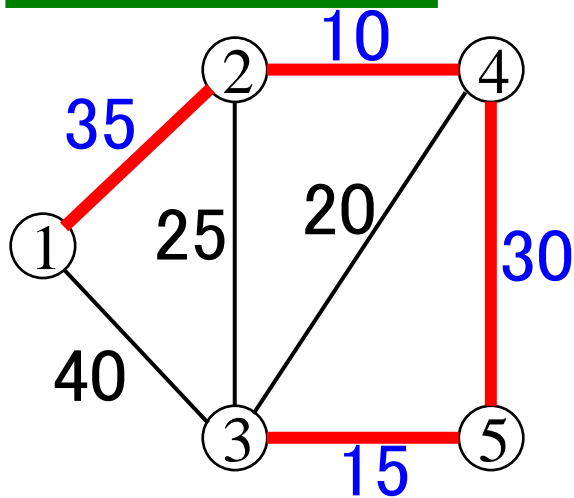
非連結部分を繋げる
○ 最小重み枝

答えが持つ性質

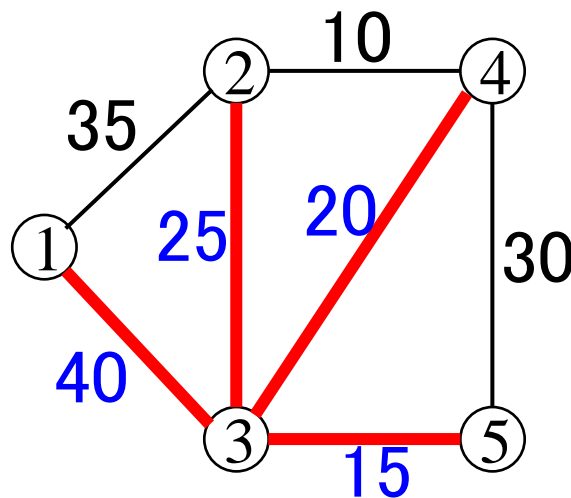
閉路は無駄 ⇒ 閉路の無いグラフ = 木
全点を結ぶ ⇒ 全張 (spanning; スパンする)

} 全張木
spanning tree

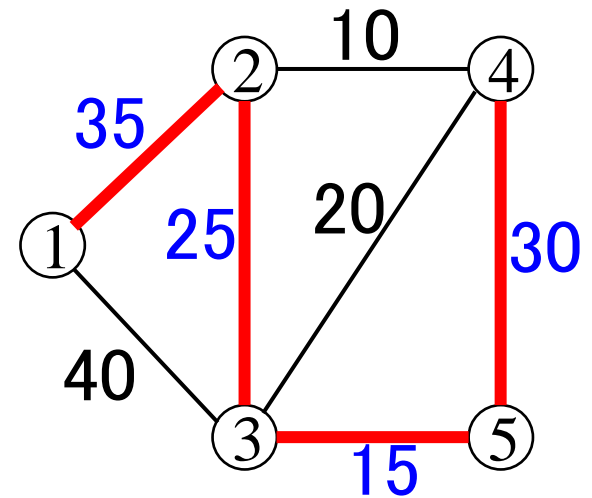
様々な全張木



$$35+10+30+15=90$$



$$40+25+20+15=100$$



$$35+25+30+15=105$$

問題の本質 重み和最小の全張木 (最小木) を見つけよ

⇔ 最小木問題

Minimum spanning tree problem

最小木の見つけ方: アイディア(1)

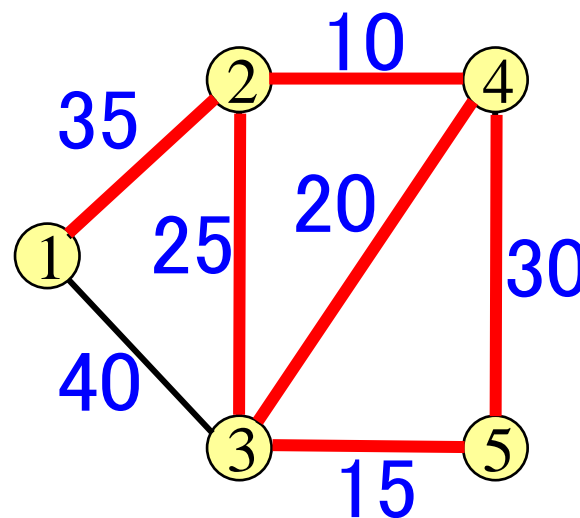
閉路⇒最大重みの枝を消去

↓ 実現方法例

重みの小さい順に枝を選択し
閉路になる時は選ばない
全点がつながったら終了

クラスカル法

(Kruskal)



最小木の見つけ方: アイディア(2)

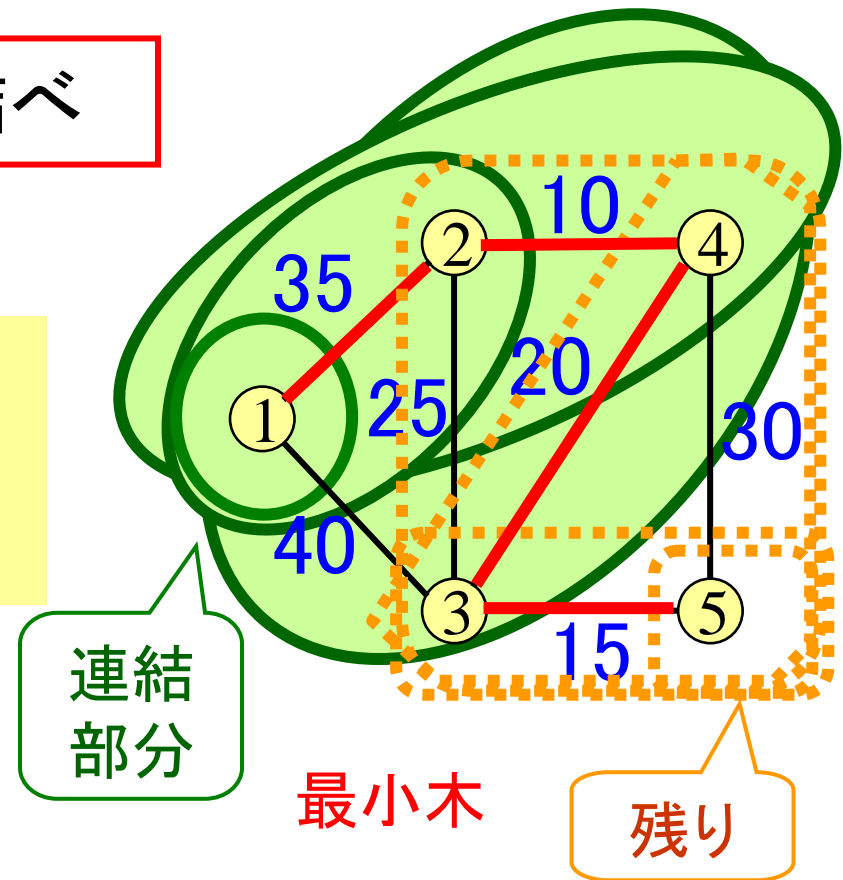
非連結⇒最小重みの枝で結べ

↓ 実現方法例

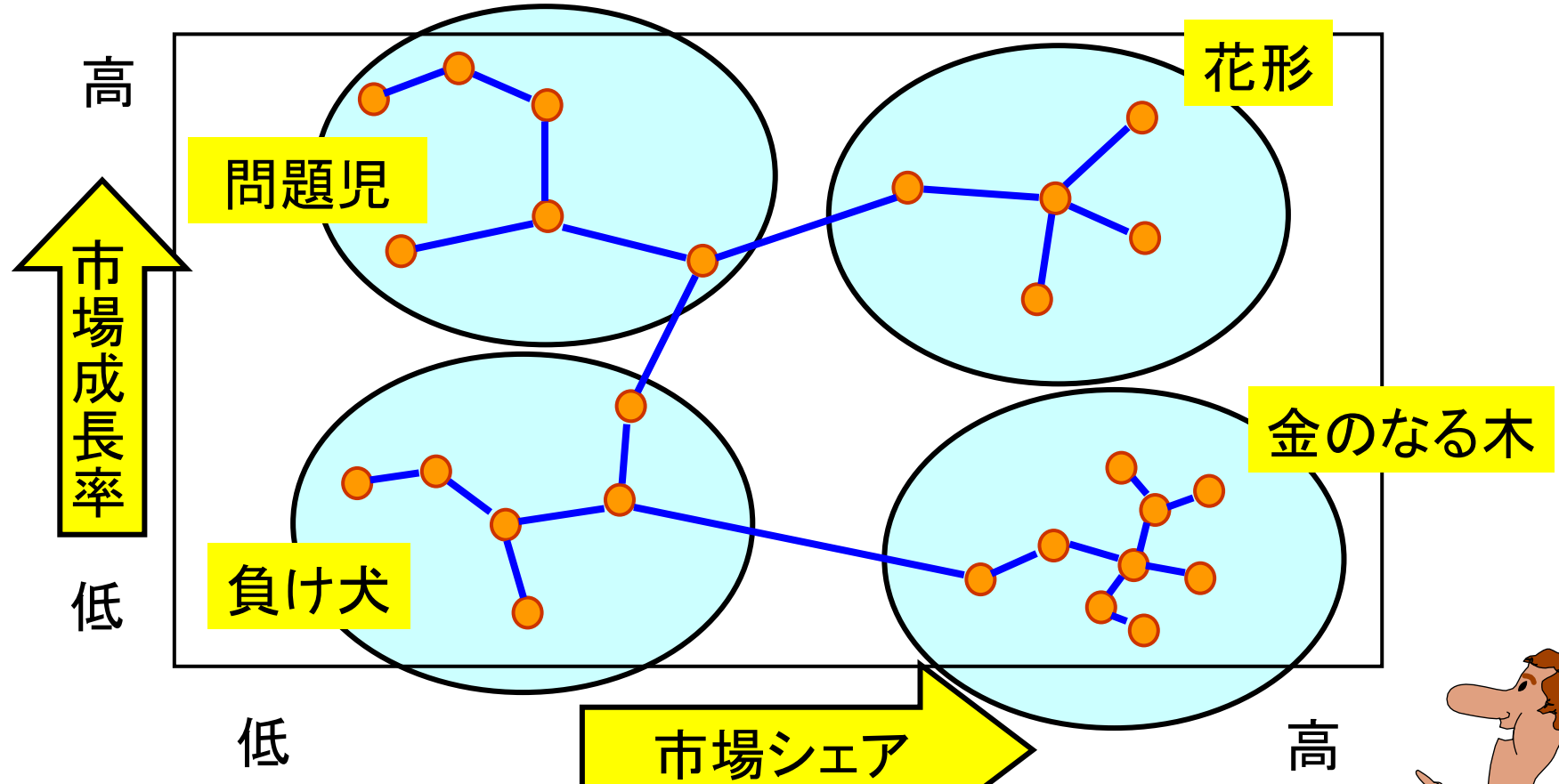
1点から連結部分を1点ずつ
最小重みの枝で増やす
全点が連結になったら終了

プリム法

(Prim)

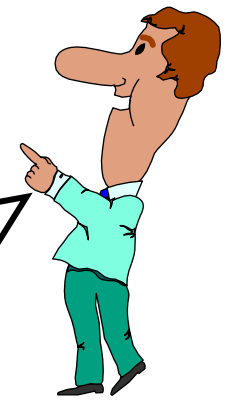


最小木問題の利用例



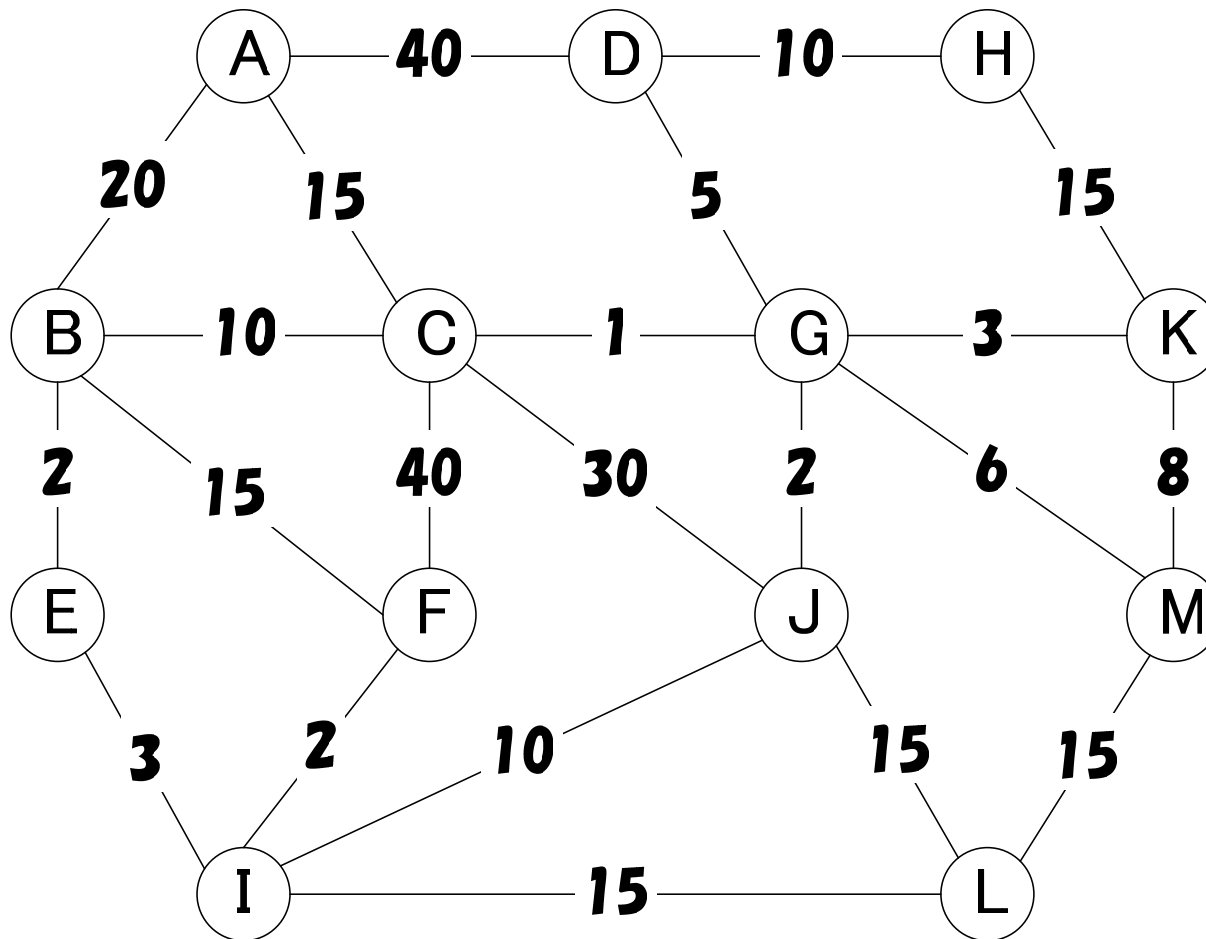
クラスター分析

4つに
分類して



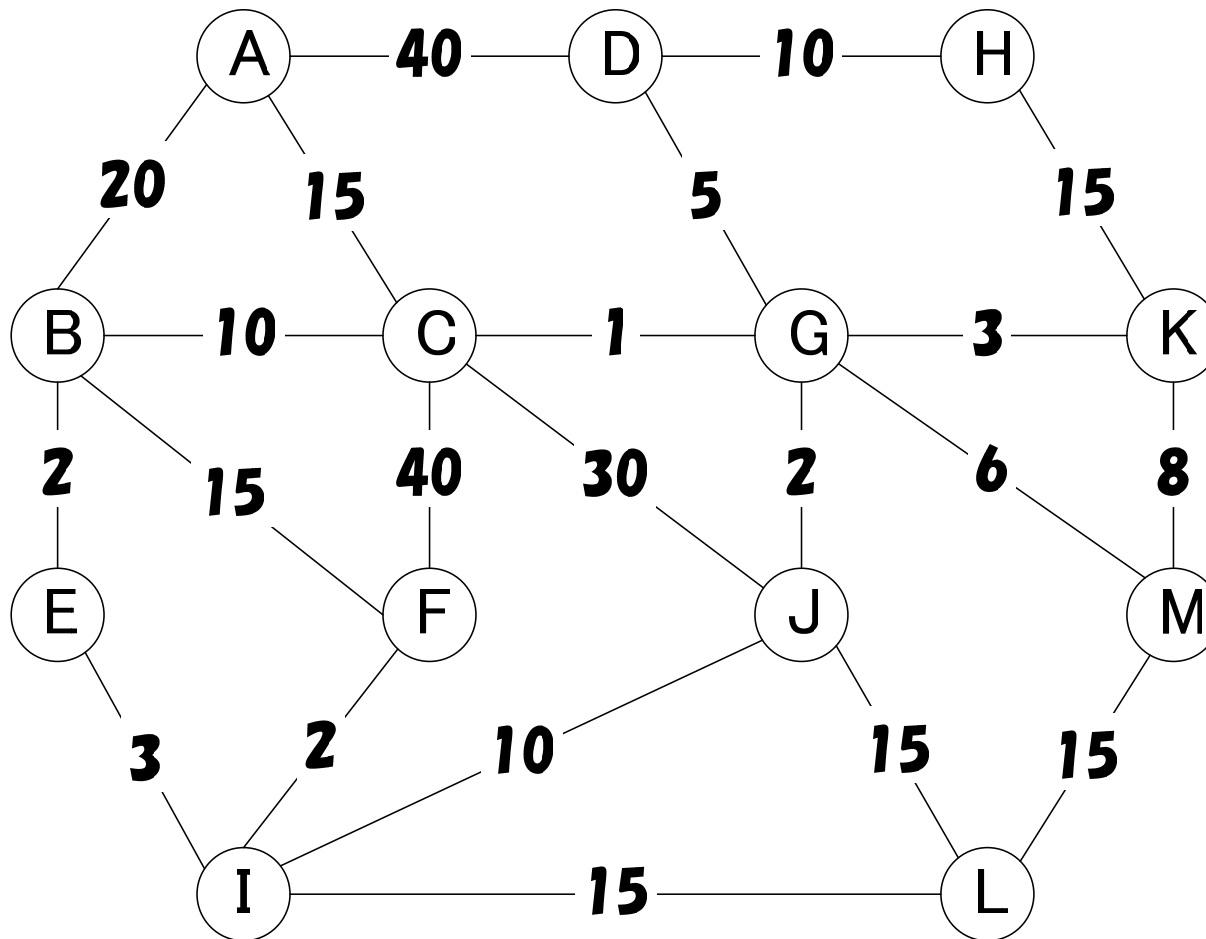
練習1

クラスカル法で**最小木**を求めよ



練習2

プリム法で**最小木**を求めよ

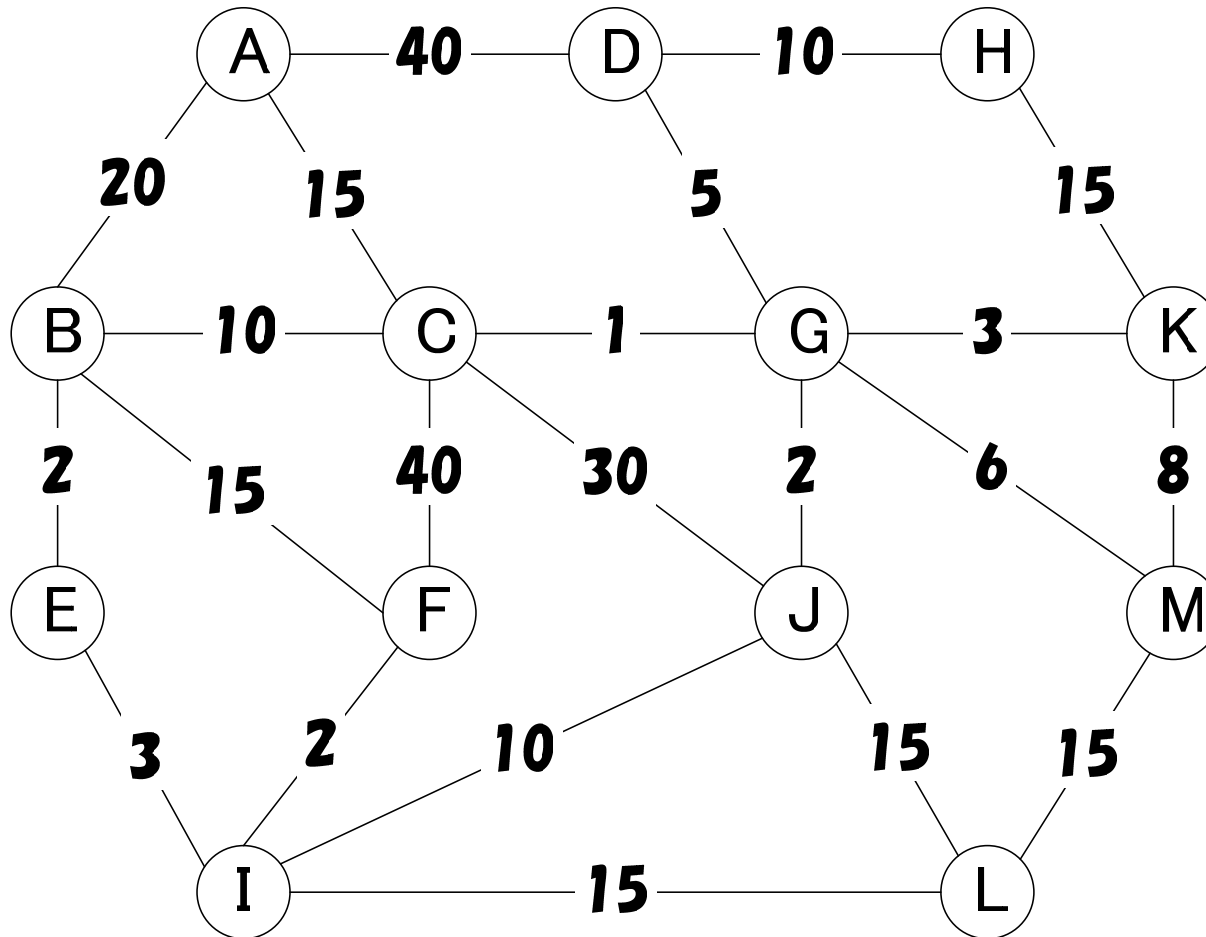


練習3

重み和が最大の全張木

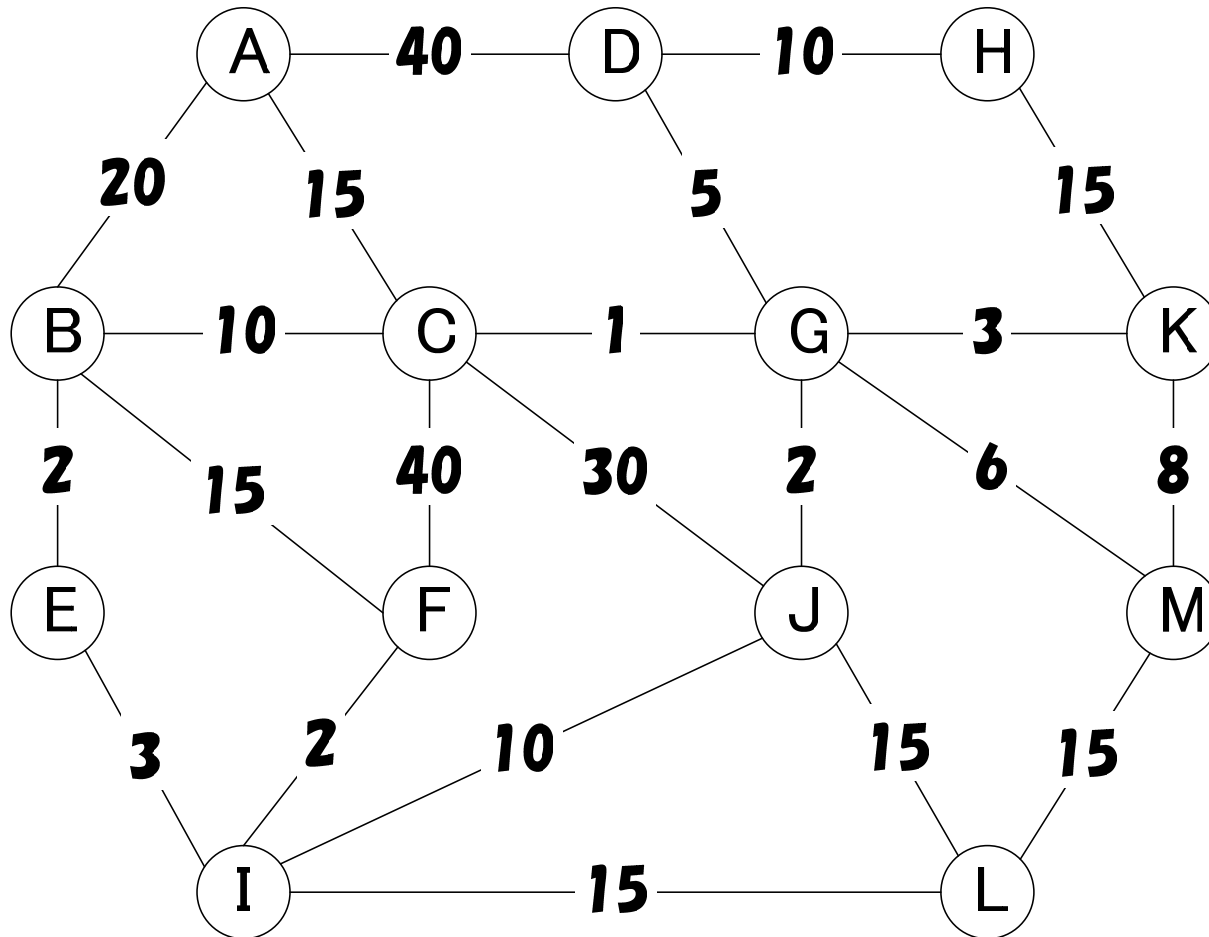


クラスカル法で**最大木**を求めよ



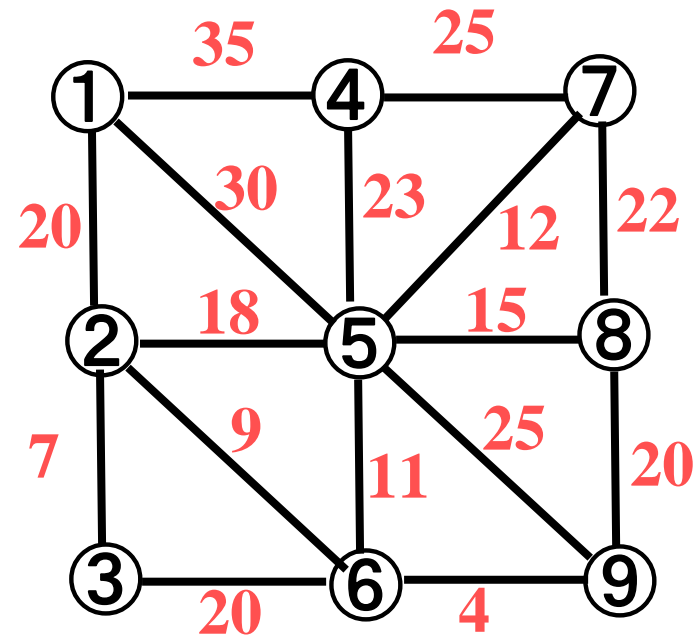
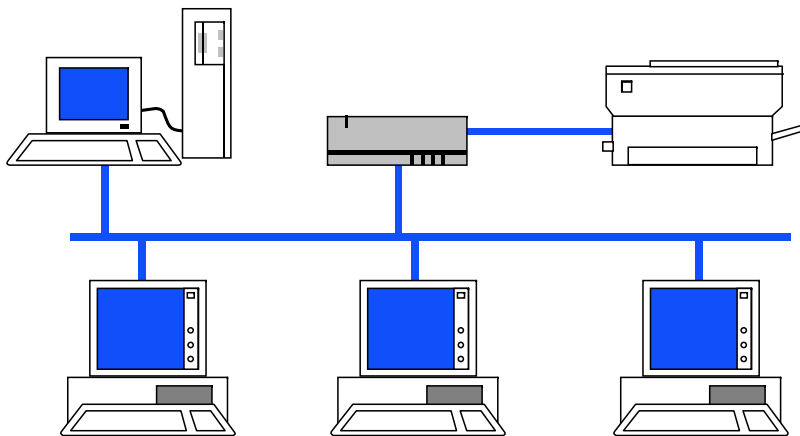
練習4

プリム法で**最大木**を求めよ



演習5-1 文教中LAN設置計画

文教中にLANを敷設することになった。コスト面から設置に用いるケーブル総延長を短くしたい。どこにLANケーブルを設置すれば良いか提案せよ。



点: 部屋
枝: 設置可能路線
数字: 設置に必要なケーブル長

演習5-2 Arc additions and deletions

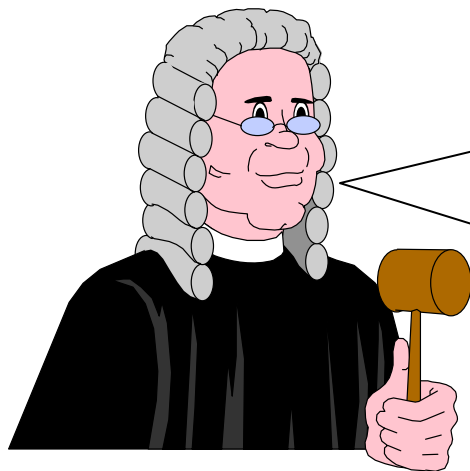
あるネットワーク上で最小木 T^* が得られている。

- (1) ネットワーク上のある枝 (i,j) が取り外された。
新たに最初から最小木を再計算する方法以外に最小木を再構築する方法を提案せよ。
- (2) ネットワーク上に重み c_{ij} を持つ枝 (i,j) が付加された。
上記同様, 最小木の再構築方法を提案せよ。

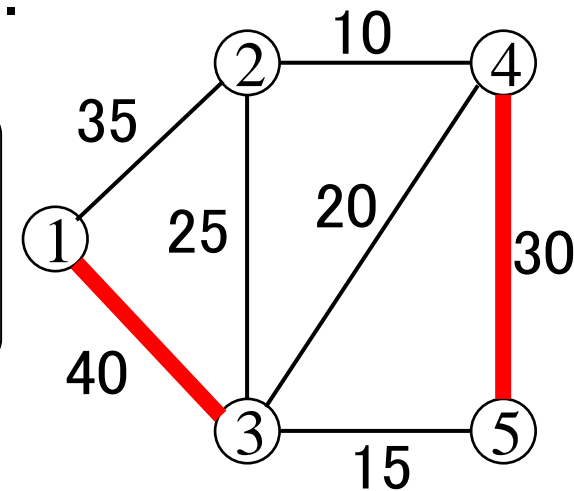
ヒント: 今まで有していた最小木 T^* の情報を有効に活かそう

演習5-3 Spanning tree containing specific arcs

- あるネットワーク上に閉路を作っていない特別な枝が何本か指定されている.
- それらの枝をすべて含む最小木を求めたい. 適切な解法を提案せよ.



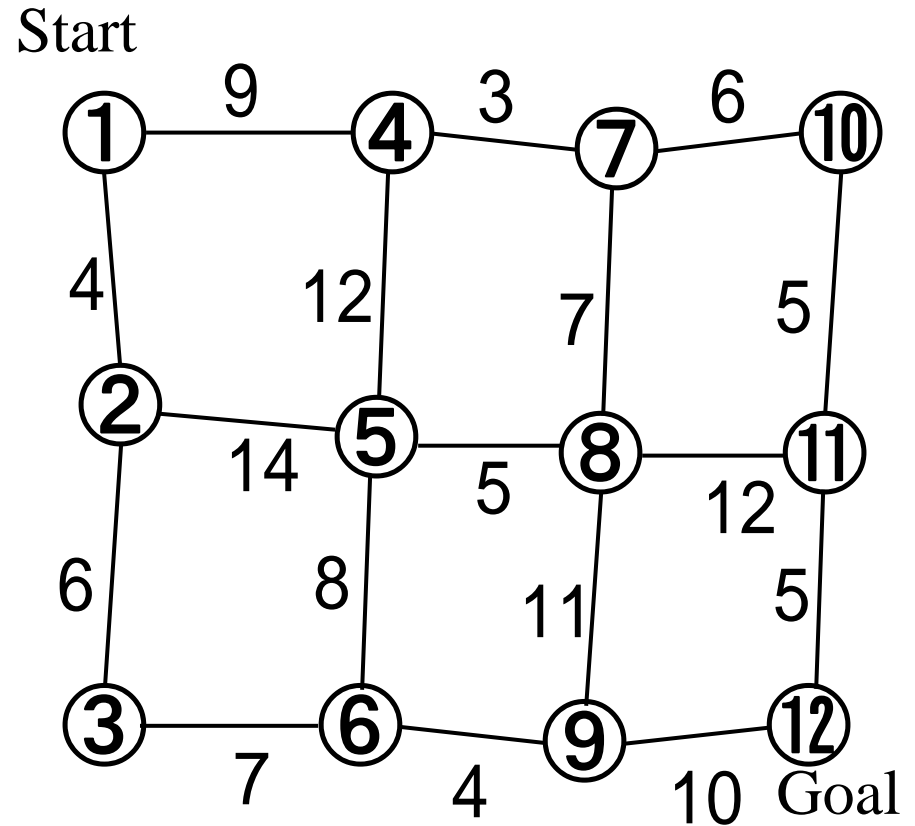
例えば, 右のネットワークで太い枝を含む最小木はどのように求められるかな?



演習5-4 砂漠横断

右の図のようなオアシスが点在する砂漠を横断する。オアシス間の最大距離が最短の道を使いたい。どこを通っていく？

(Minimax path 問題)



点:オアシス, 枝:道, 数字:距離