D: 一日乗車券

原案：井上
問題：伊藤
解答：伊藤・井上・森・保坂

問題概要

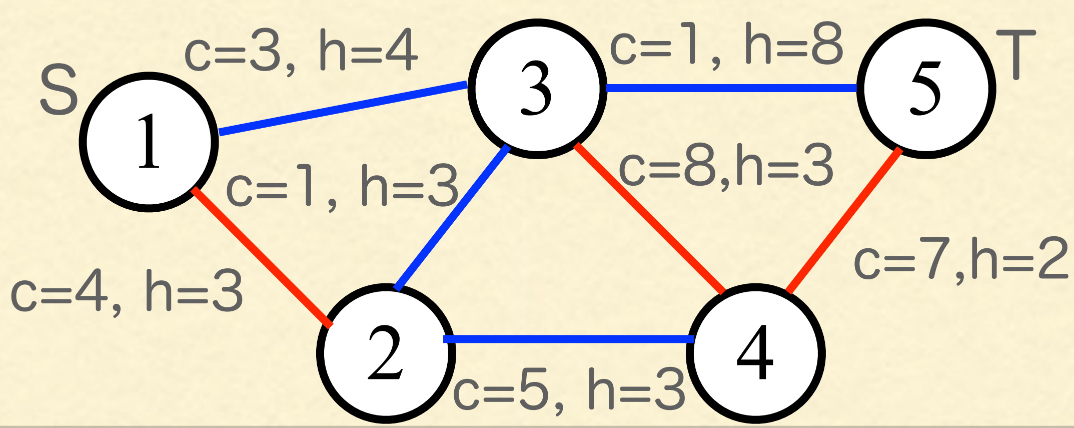
- N駅M路線からなる鉄道網がある
- 各路線はK個ある会社のいずれか1つに管理されている
- P種類の1Dayパスポートが販売されている
  - 記載された(複数の)会社が管理する路線がすべて無料
- 駅Sから駅TへH時間以内に到達する最小費用を求めよ

- 制約: 2≦N≦100, 1≦M≦500, 1≦K≦8, 1≦H≦24

例: H =10, P = 2

・d=10, {A社}

・d=15, {A社,B社}

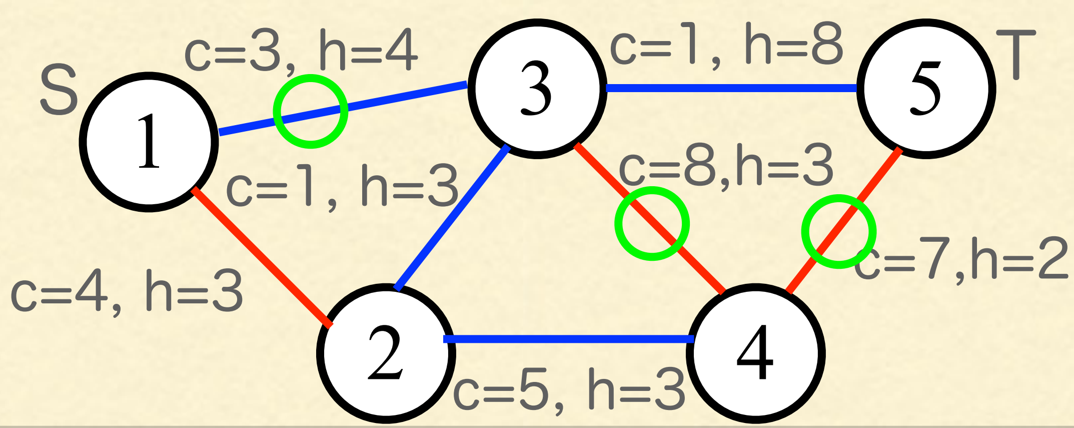


問題概要

- N駅M路線からなる鉄道網がある
- 各路線はK個ある会社のいずれか1つに管理されている
- P種類の1Dayパスポートが販売されている
  - 記載された(複数の)会社が管理する路線がすべて無料
- 駅Sから駅TへH時間以内に到達する最小費用を求めよ

- 制約: 2≦N≦100, 1≦M≦500, 1≦K≦8, 1≦H≦24

例: H =10, P = 2
- d=10, {A社}
- d=15, {A社,B社}



$\boxed{\text{最小費用: 13}}$

問題概要

- N駅M路線からなる鉄道網がある
- 各路線はK個ある会社のいずれか1つに管理されている
- P種類の1Dayパスポートが販売されている
  - 記載された(複数の)会社が管理する路線がすべて無料
- 駅Sから駅TへH時間以内に到達する最小費用を求めよ

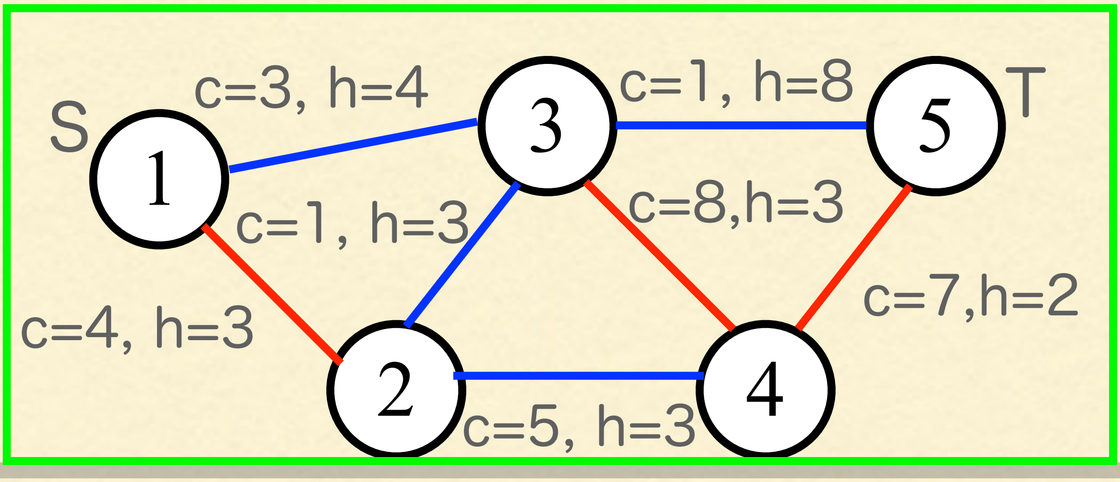
グラフの　経路の　最小　= ダイクストラ！！

- 制約: 2≦N≦100, 1≦M≦500, 1≦K≦8, 1≦H≦24

例: H =10, P = 2

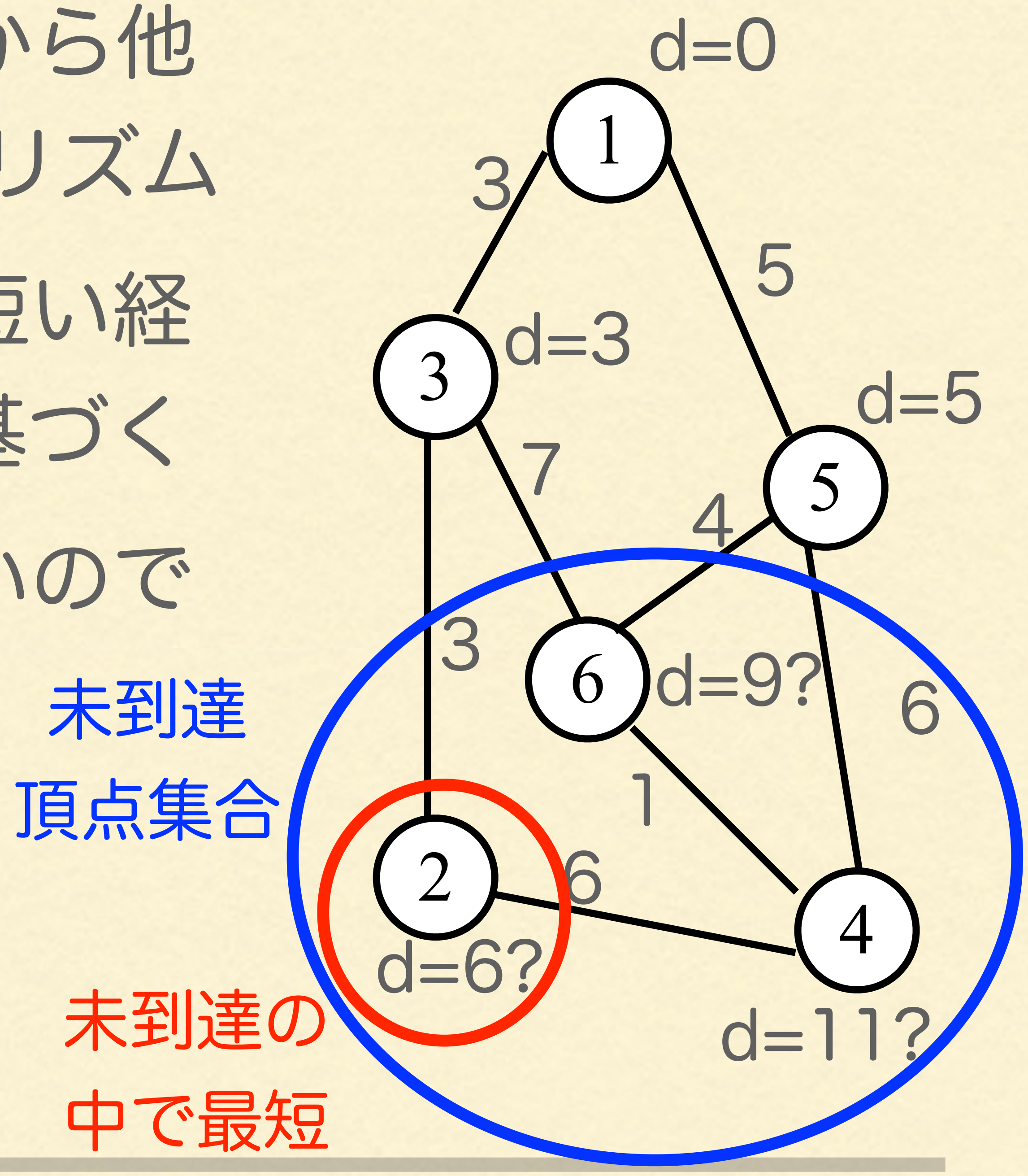
・d=10, {A社}

・d=15, {A社,B社}



ダイクストラ法

- グラフの単一始点最短路問題(1点から他の全頂点への最短路)を解くアルゴリズム
  - 未到達頂点への経路の内、最も短い経路を選択し続ける貪欲な戦略に基づく
  - 負の辺があると正しく動作しないので注意！！！ (今回は大丈夫)
- $O(N^2)$の実装と2分ヒープを用いた$O(M\log N)$の実装がよく用いられる



ダイクストラ法？

- N駅M路線からなる鉄道網がある
- 各路線はK個ある会社のいずれか
- P種類の1Dayパスポートが販売されている
  - 記載された(複数の)会社が管理する路線がすべて無料
- 駅Sから駅TへH時間以内に到達する最小費用を求めよ

問題点2:辺のコストが変化

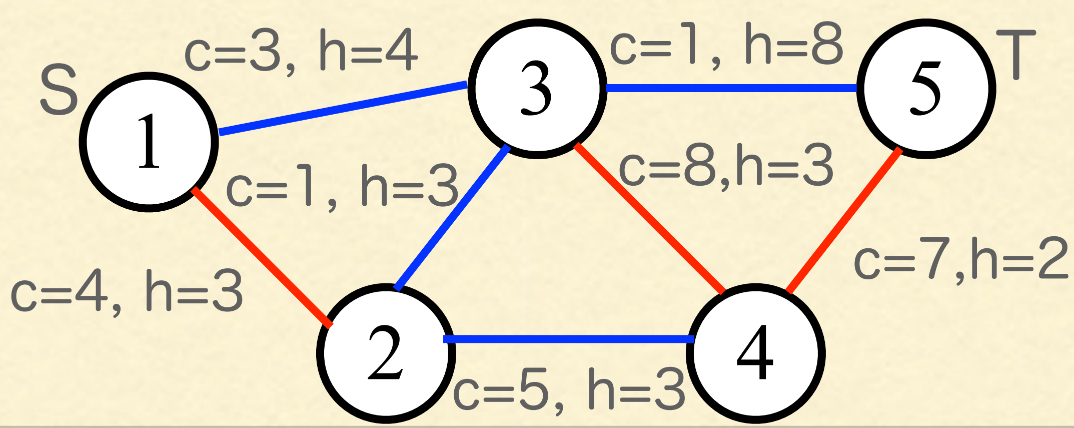
問題点1:制限時間がある

- 制約: 2≦N≦100, 1≦M≦500, 1≦K≦8, 1≦H≦24

例: H =10, P = 2

・d=10, {A社}

・d=15, {A社,B社}

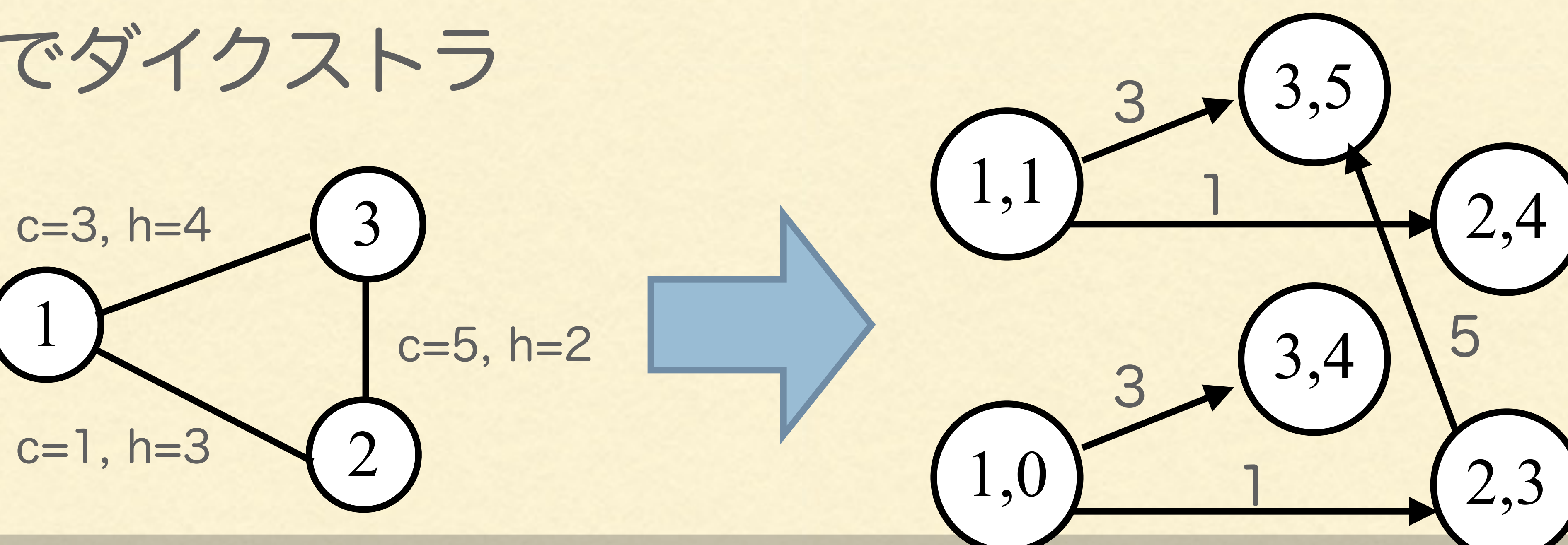


解決策1: グラフ拡張

- 今，Sから出発して，頂点vに時刻hで辿り着いたとする．
- これまでのコストに，vからTへ(H-h)時間以内に辿り着ける最短路を足したものが，SからTへの最短路候補の1つ
- 逆に言えば，頂点vに時刻hで辿り着くような経路のうち，最短路のみを考えればよい → 組(v,h)を頂点とした拡張グラフでダイクストラ

頂点数:N

辺数: M



頂点数:HN

辺数: HM

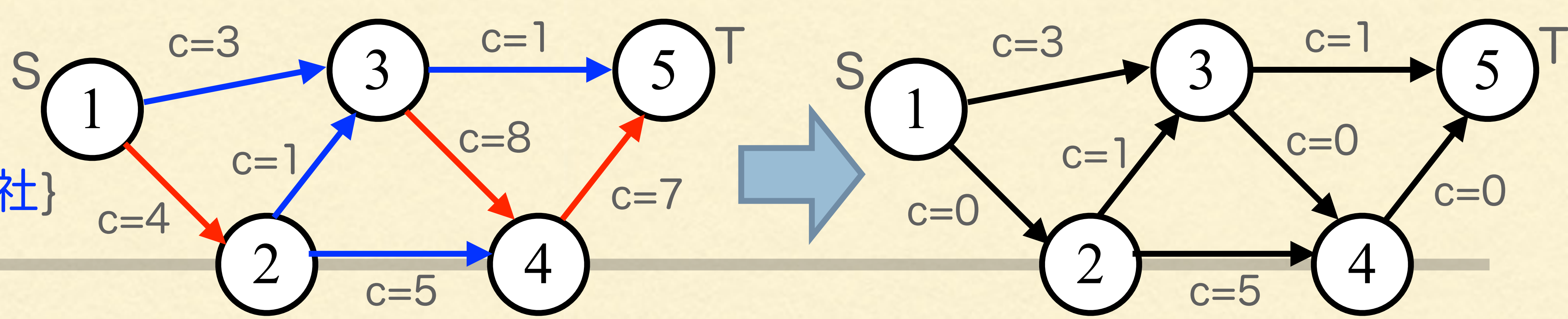
解決策2: 全探索

- あらかじめ使うパスポートを決め打ちすると，パスポートの総額も，無料になる路線もわかる
  - 元のコストの辺とコスト0の辺からなるグラフが作れる
  - このグラフの上ならダイクストラが使える
- すべてのパスポートの組み合わせに対して，「パスポートの総額+ダイクストラで求めた最小コスト」の最小が解となる

P = 2

・d=10, {A社}

・d=15, {A社,B社}



解決策2: 全探索+動的計画法

- しかし，パスポートの種類Pは最大$2^K-1=255$個あるので，すべての組み合わせは最大$2^{255}≒10^{76}$で膨大→全探索無理
- 重要なのは「どの会社が無料になるか」
  - 会社の組み合わせは最大$2^K=256$通り → 全探索余裕
  - ある会社の組み合わせを無料にするようにパスポートの組を選ぶとき，パスポートの総額を最小にする買い方のみをすればよい → 動的計画法で求まる

動的計画法

- Dynamic Programming (DP)とも呼ばれる
- 漸化式を立て，その漸化式を小さい方から計算していく
- 今回の漸化式は,
  - $DP[V] = \min\{ DP[V'] + d_i \mid V = V' \cup P_i\}$
  - V, V': 会社集合, d_i: i番目のパスポートの料金
  - P_i: i番目のパスポートの会社集合 $\{k_{i,1}, ..., k_{i,l_i}\}$
- 小さいV'から順に，すべてのP_iとの和集合をとるように漸化式を更新することで，$O(P・2^K)$で計算することができる

解法まとめ

- 会社の部分集合Xすべて(2^k通り)について,
  - Xに含まれるすべての会社を無料にできるパスポート組み合わせの最小費用を(DPで)求める (=C_Xとする)
  - Xに含まれるすべての会社の路線が無料になった拡張グラフ上で最小費用を(ダイクストラで)求める (=D_Xとする)
- $C_X + D_X$の最小値が答え
- 計算量: $O(2^k \times (\ P + HM\log(HN)\ )\ )$

発展：最短路計算の高速化

- (頂点×時間)の拡張グラフは，DAG (Directed Acyclic Graph, 閉路のない有向グラフ)になっている
  - 移動にかかる時間が必ず正なので，一度移動すると同じ時間に戻ることはできない
- DAG上の最短路問題は線形時間(O(N+M))で解くことができる
  - まだ確定していない頂点から，その頂点に到達する方法がもうないところから順に(=トポロジカル順序に従って)最短路を更新
  - これにより前述の計算量がさらに落とせる
  - 今回はこれで高速化しなくても十分余裕を持って間に合います

ジャッジ解

- 伊藤: 142行, 2900B (C++)
- 井上: 111行, 2200B (C++)
- 森: 133行, 3200B (C++)
- 保坂: 95行, 2400B (Java)

解答状況

- First Acceptance:
  - !#$%&()*+-./:;<=>?@[\]^_`{|}~ (00:40:30)
- AC / Submission
  - 40 / 48 (83.3%)
- AC / Trying Teams
  - 40 / 42 (95.2%)