2014/6/9 Mon.

問題解決技法入門

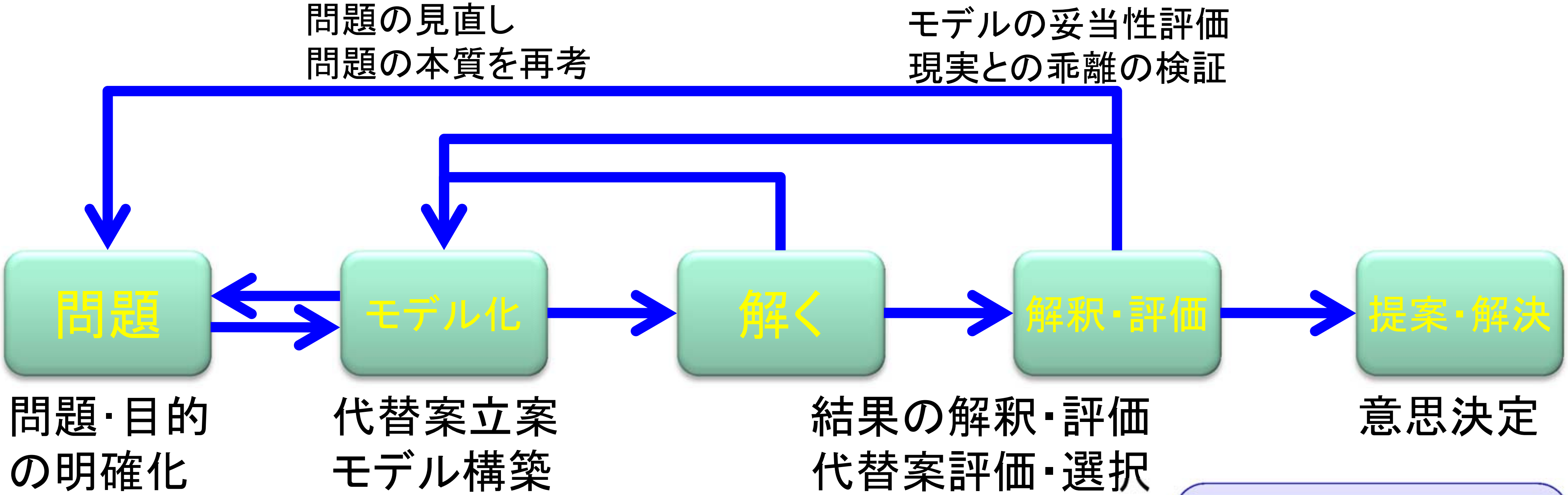
最短経路探索

堀田 敬介

意思決定

論理的思考力
データ分析，統計学
数理的アプローチ

- 「問題の把握」から「意思決定」までの流れ



- 問題発見・状況認識
  - 状況を把握し問題の背後にある本質を追究
  - いったい何を知りたいのか？
  - 問題の本質は何か？

現状認識力
問題発見・定義

- 推論・モデル作成
  - 推論に基づきモデル作成
  - 現実を支配する法則を数量的に明確化

- 答えを導く
  - 解法選択
  - 解法構築
  - パラメータ調整

説得力
問題解決力

- 結果評価・解釈
  - 解法のもたらす結果の解釈・考察
  - 得られた代替案の評価・分析

カー・ナビゲーションとは？

航海，航海術

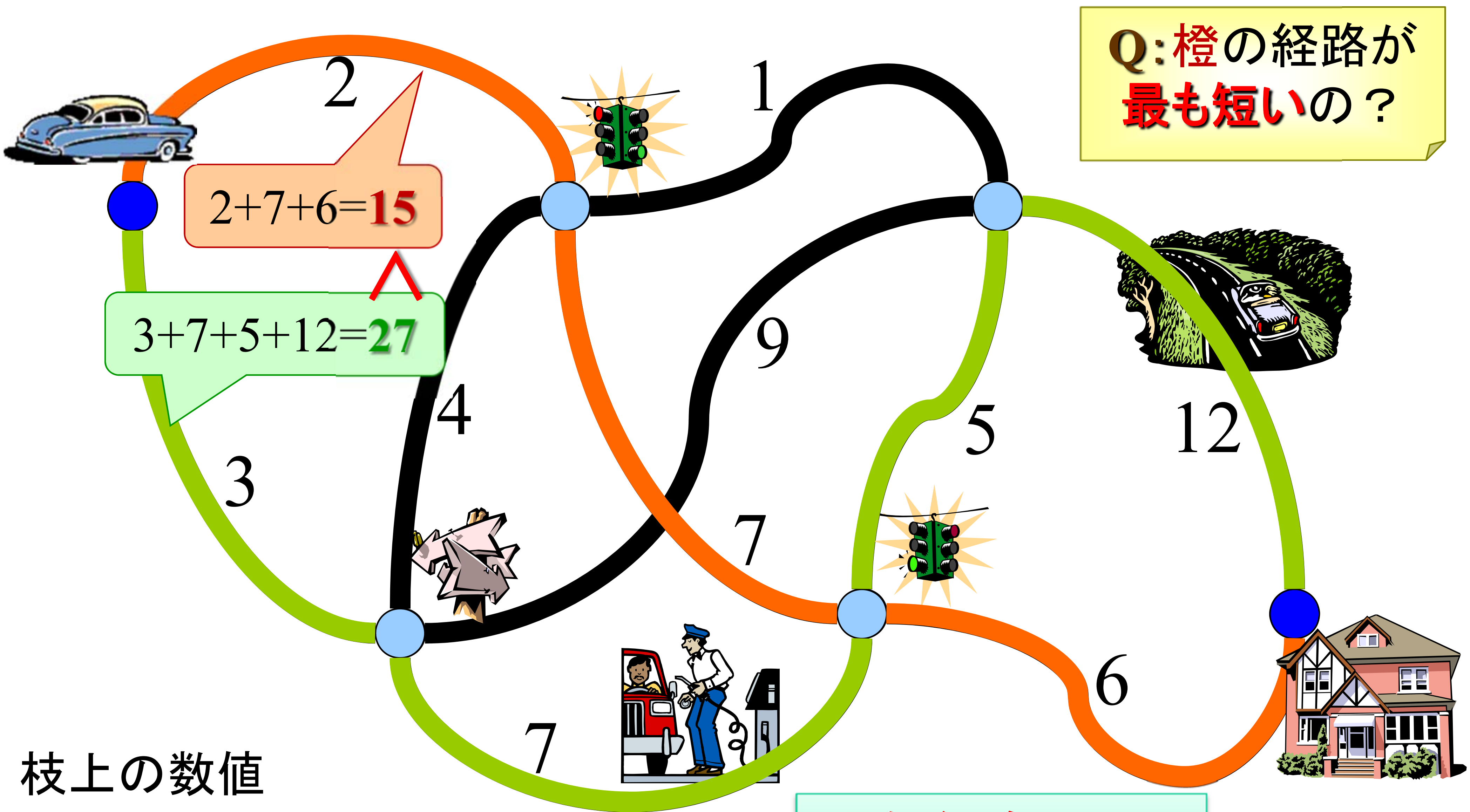
PND・GPS携帯・スマホなど多様

- カー・ナビゲーション・システム
  automotive navigation system

- ナビゲーション・システムの2大機能
  - （リアルタイム）情報表示
    - 現在地や渋滞情報，周辺情報などを地図に重ねて表示
  - ルート探索
    - 目的地を指定すると現在地からの（最短）経路を表示

どうやってるの？

最短経路はどこにあるのか？



枝上の数値
＝コスト
＝距離，時間，費用，etc.

最短経路を見つけたいのダカラ！

最短経路はどこ？

オレンジの経路が最短なのか？
どうすればわかる？

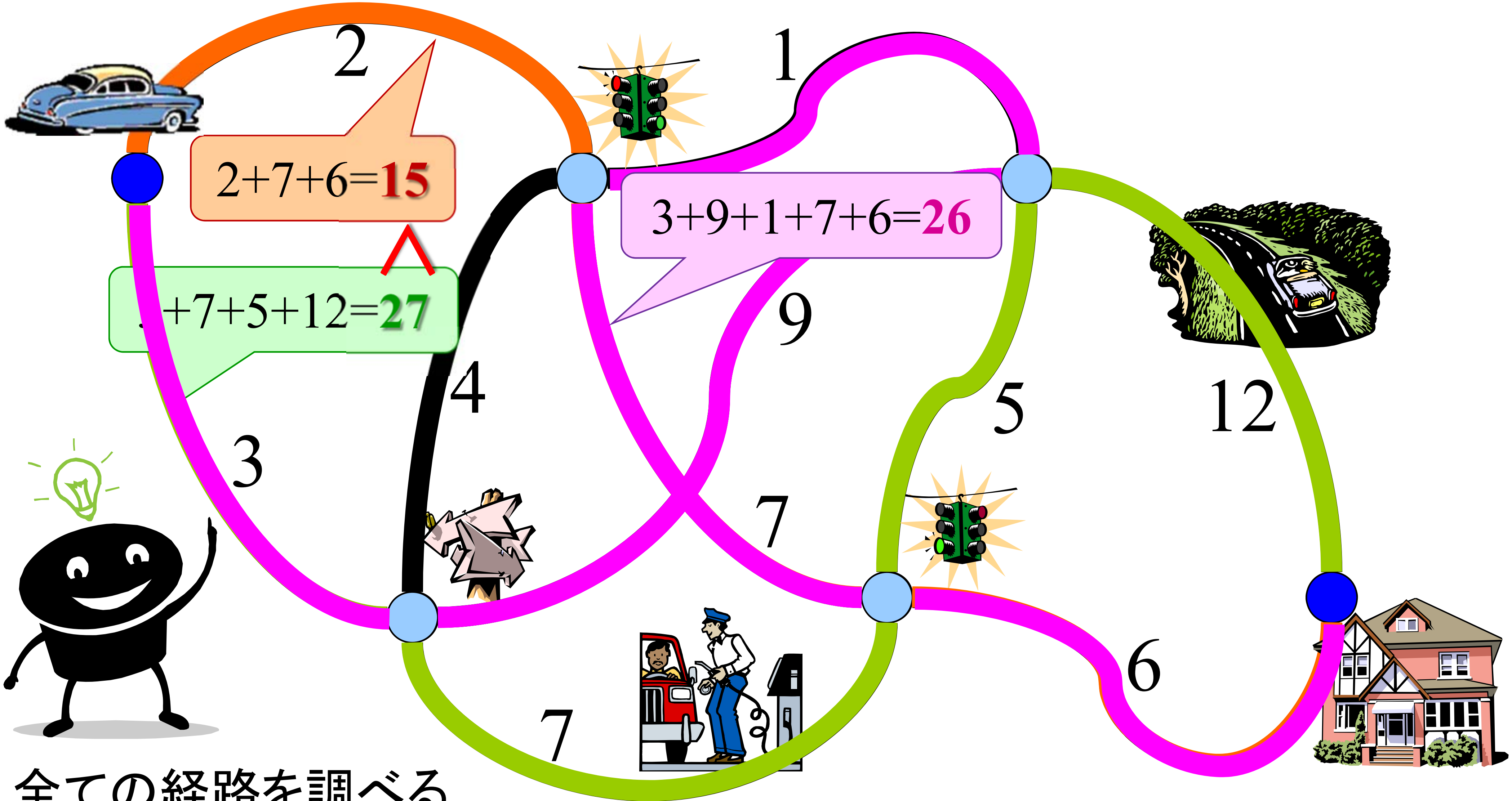
全ての経路を調べ，その中から

最も短い経路を選べば良い！

〔素朴で素直な方法〕

＝全列挙，しらみつぶし

しらみつぶし！



全ての経路を調べる

- ✓ 交差点は2度通らない
- ✓ 同じ道路は2度通らない

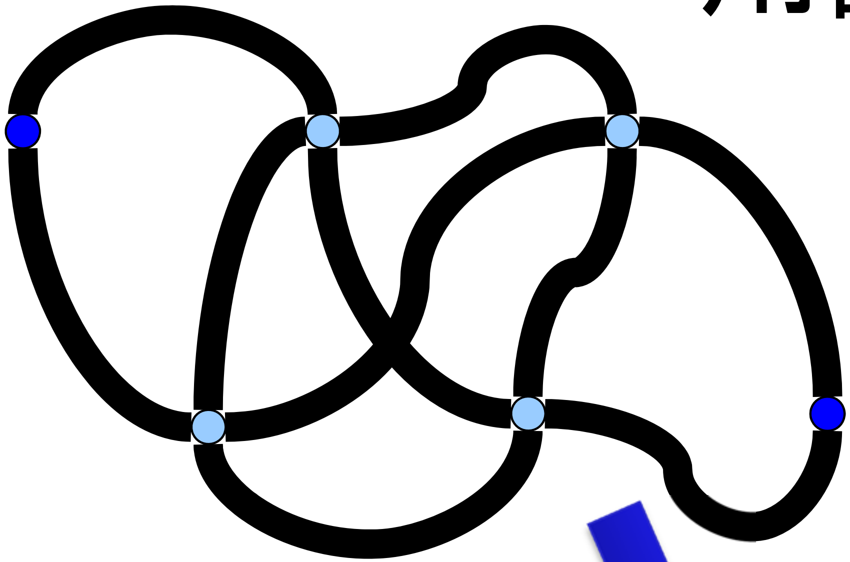
➡ 最短経路がわかるよね！

次に進む前に...

用語の説明

グラフ理論
graph theory

グラフ	ネットワーク
graph	network

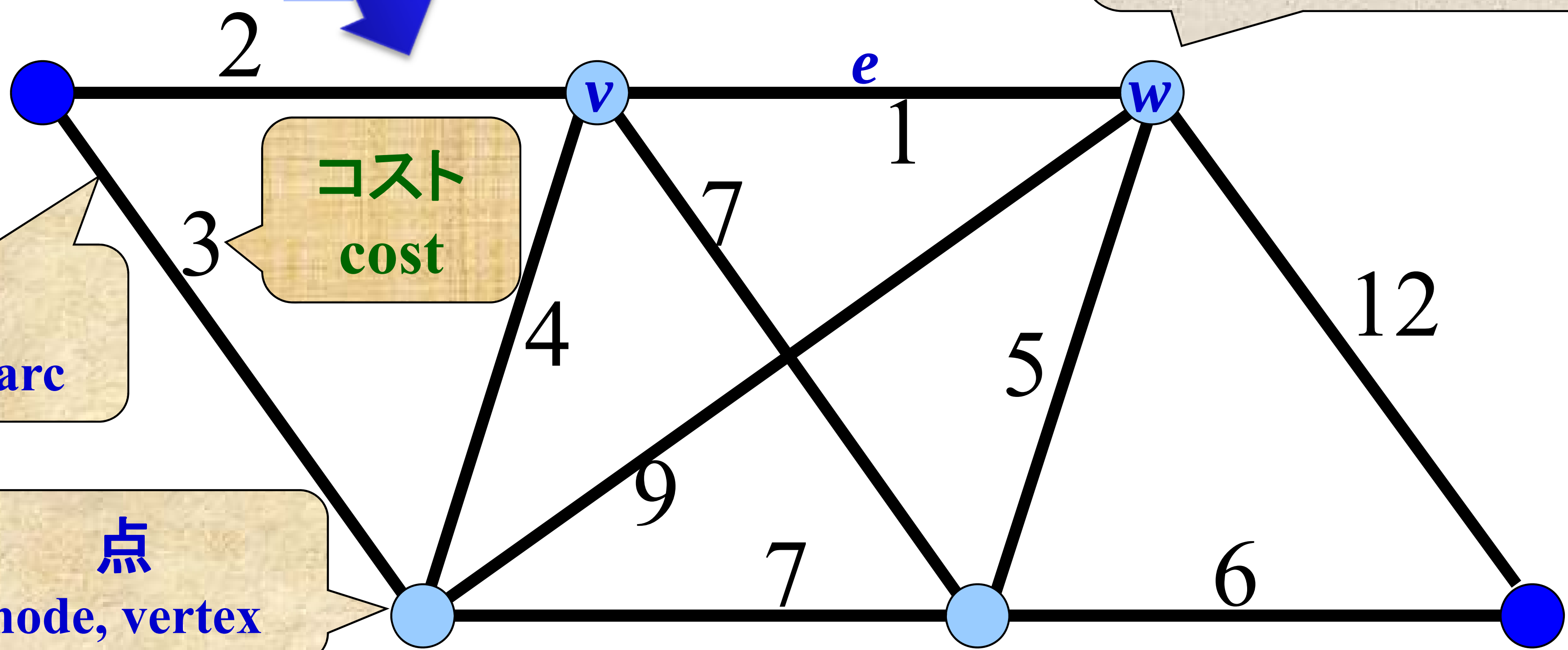


現実の経路を抽象化

隣接・接続関係

v **and** ***w*** **are adjacent.**

v **is incident to** ***e*.**



コスト
cost

枝
edge, arc

点
node, vertex

次に進む前に…

何故わざわざグラフにして考えるのか？

- 必要な情報を簡潔に過不足なく表現できる
  – 点の数
  – 枝の数
  – 点と枝の接続関係
  （点と点の隣接関係）
  – 枝のコスト

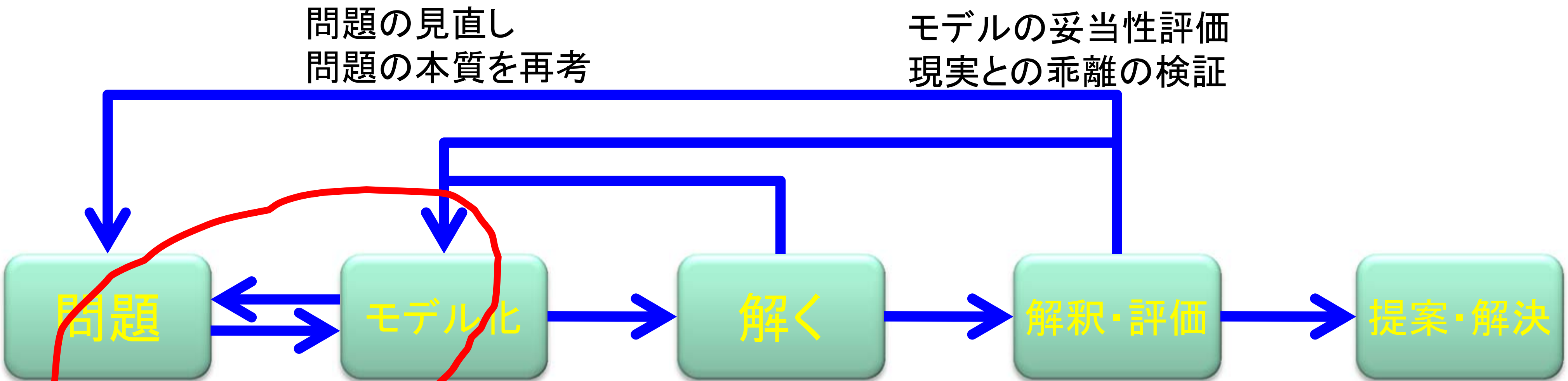
さらに…
グラフ理論で培われた豊かな知恵を利用できる

- (ここでは) 必要ない情報
  – 路の途中に上り下りなど坂道がある,
  – 路の途中にカーブが何回あるか,
  – 点や枝の正確な位置, etc.

意思決定

論理的思考力
データ分析，統計学
数理的アプローチ

- 「問題の把握」から「意思決定」までの流れ



問題・目的の明確化

代替案立案
モデル構築

結果の解釈・評価
代替案評価・選択

意思決定

- 問題発見・状況認識
  - 状況を把握し問題の背後にある本質を追究
  - いったい何を知りたいのか？
  - 問題の本質は何か？

- 答えを導く
  - 解法選択
  - 解法構築
  - パラメータ調整

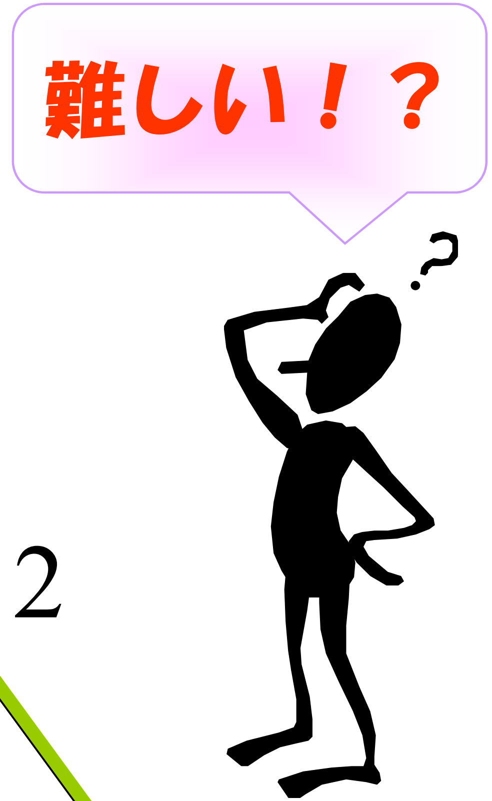
説得力
問題解決力

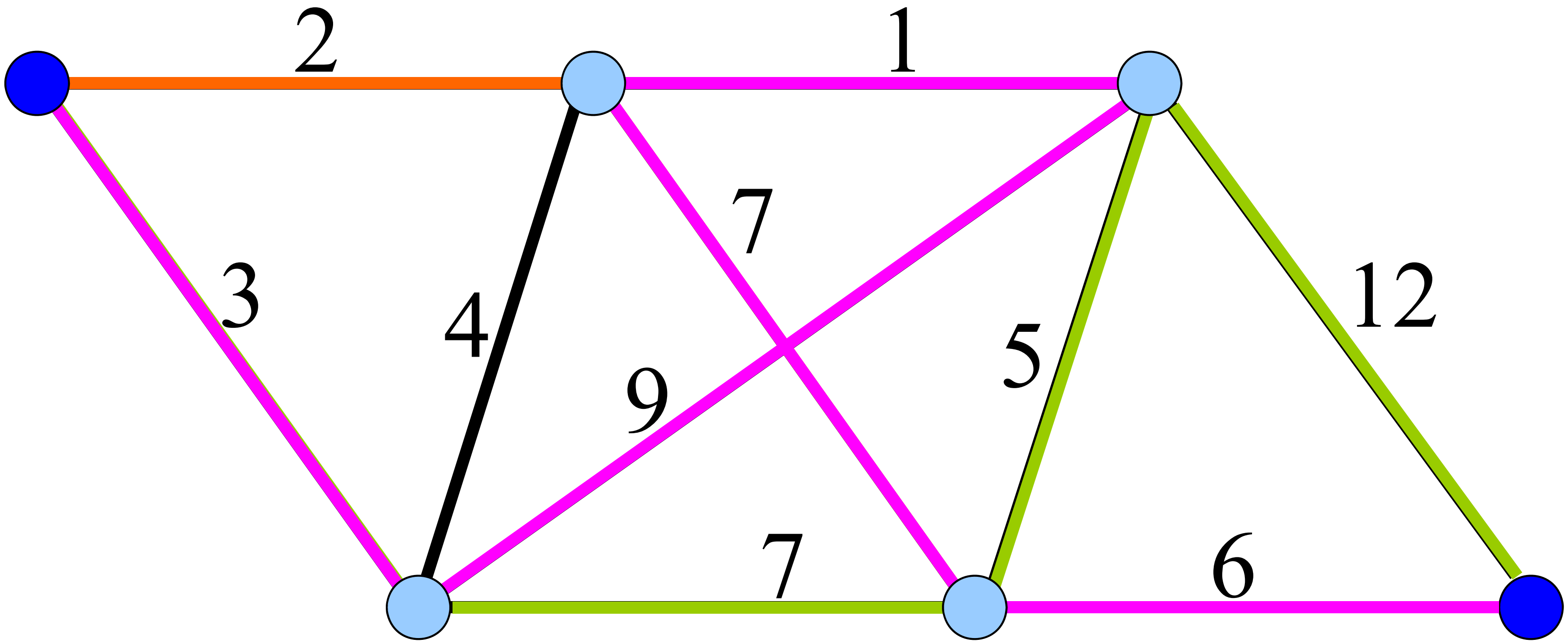
- 結果評価・解釈
  - 解法のもたらす結果の解釈・考察
  - 得られた代替案の評価・分析

- 推論・モデル作成
  - 推論に基づきモデル作成
  - 現実を支配する法則を数量的に明確化

現状認識力
問題発見・定義

全ての経路を探す！？



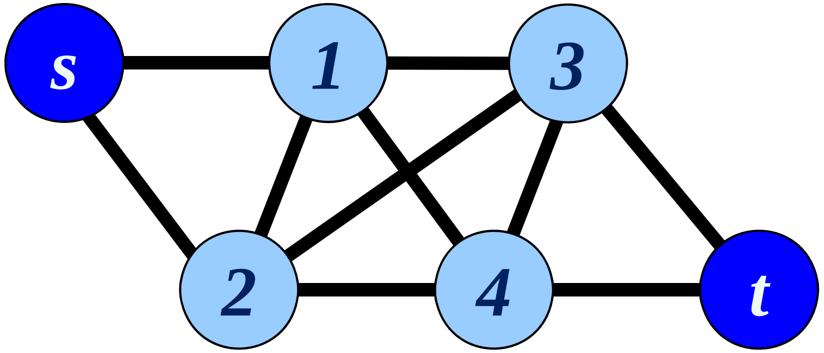


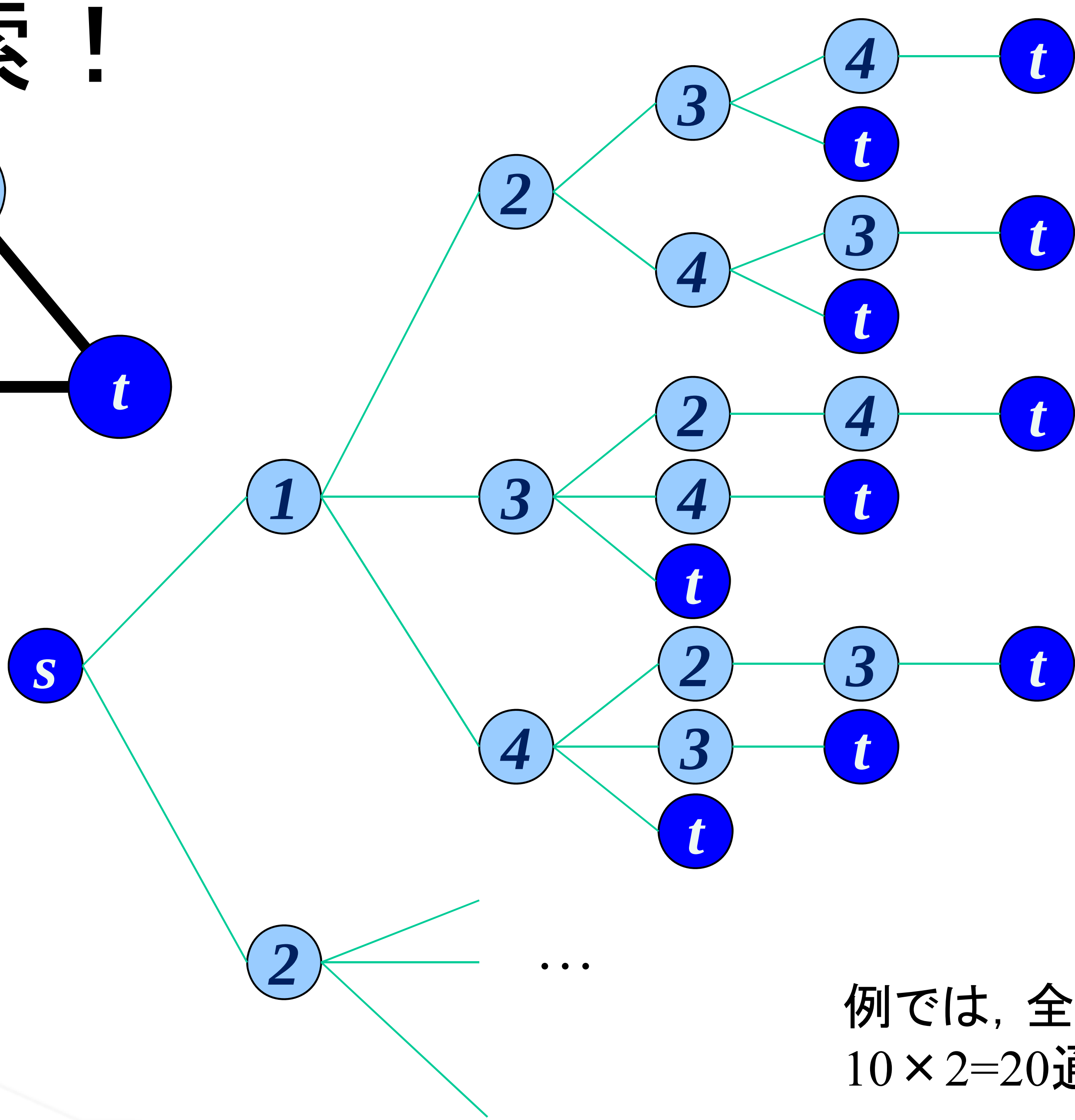
一体何通りの経路があるんだろう？
どうやって数えたらいいの？

全ての経路を調べる

- ✓ 交差点は2度通らない
- ✓ 同じ道路は2度通らない

<u>全</u>経路探索！





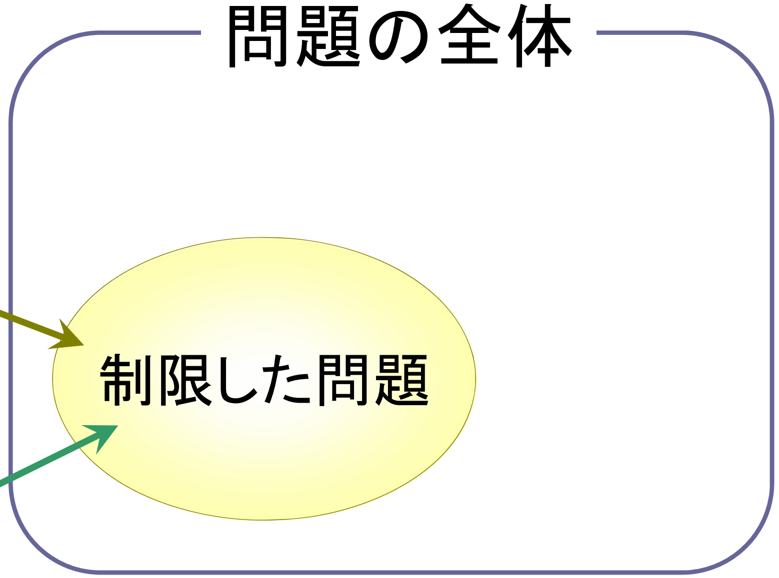
例では，全経路
10×2=20通り

注）グラフに閉路（cycle）がなければ，
動的計画法（DP）で数え上げは可能

難しいなら易しくすればいいのさ！

OR的問題解決のヒント
問題を簡単にする！

- 問題の一部だけを考える
- 条件を付加して易しくする



ここだけで考えて上手くいけば，
全体に広げられるかも！

全てのネットワーク上の最短路問題

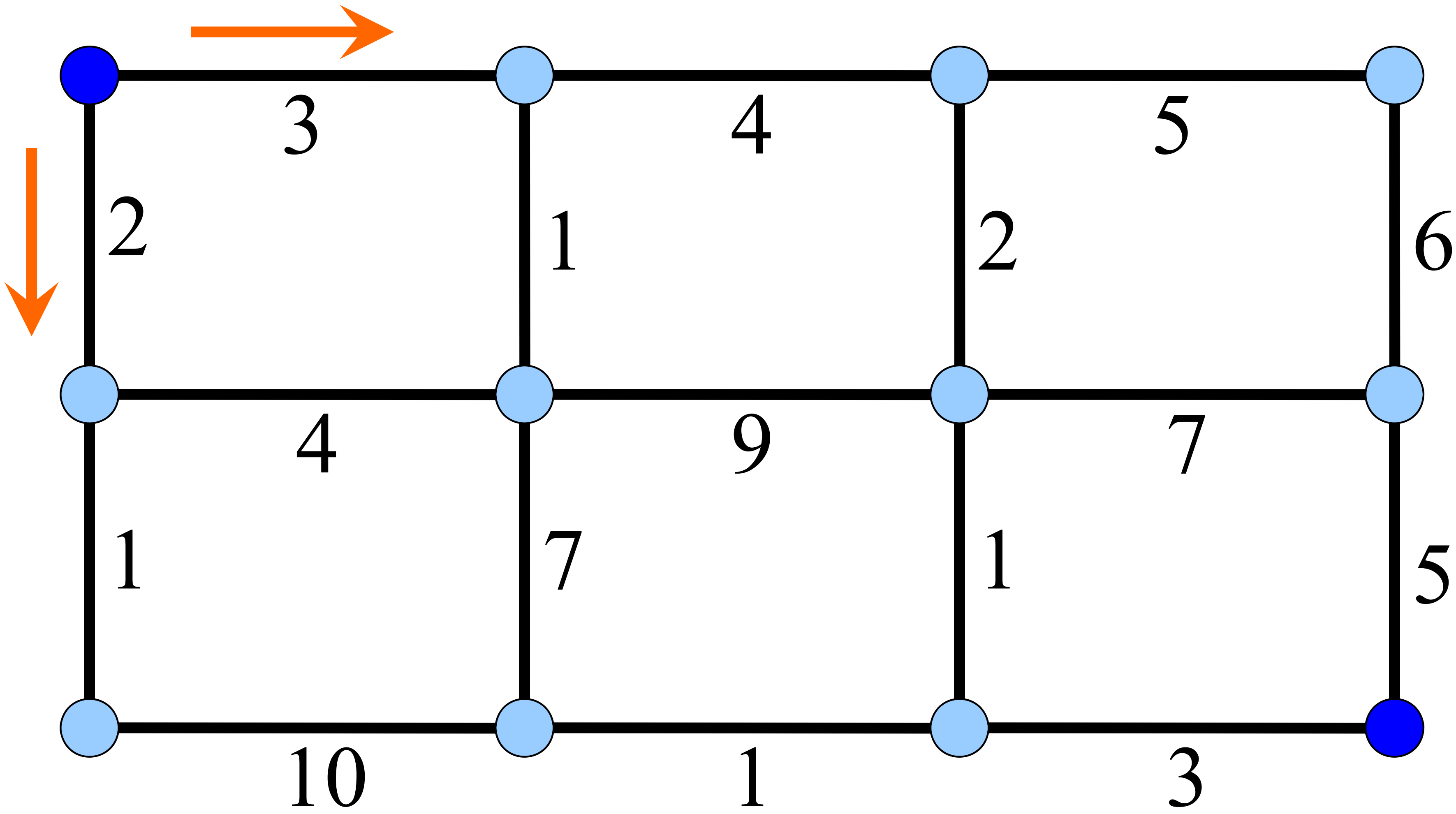
制限した問題

- 格子状のネットワーク
- 出発地：左上点，目的地：右下点
- 移動は右・下方向へのみ

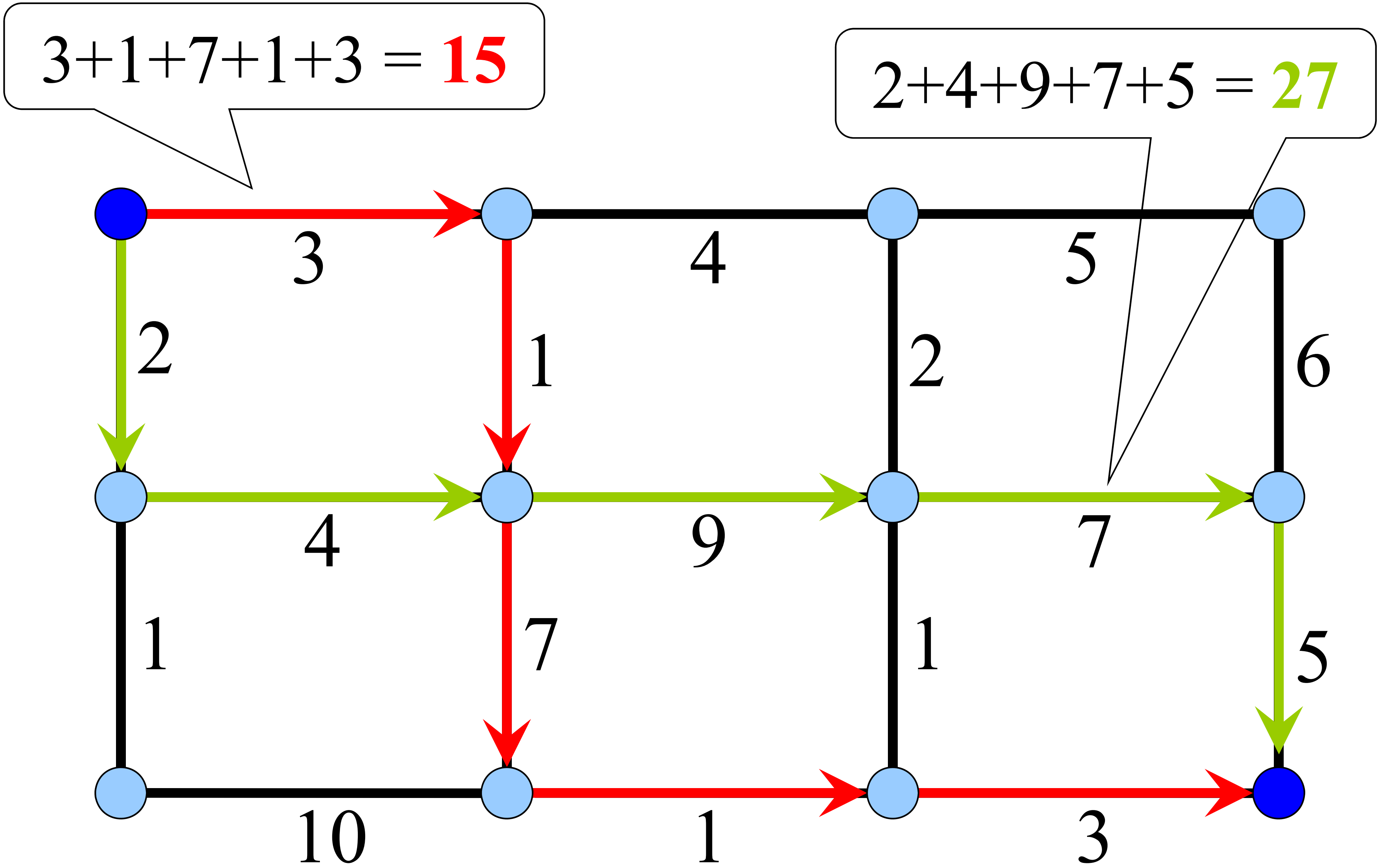
難しいなら易しくすればいいのさ！

制限した問題

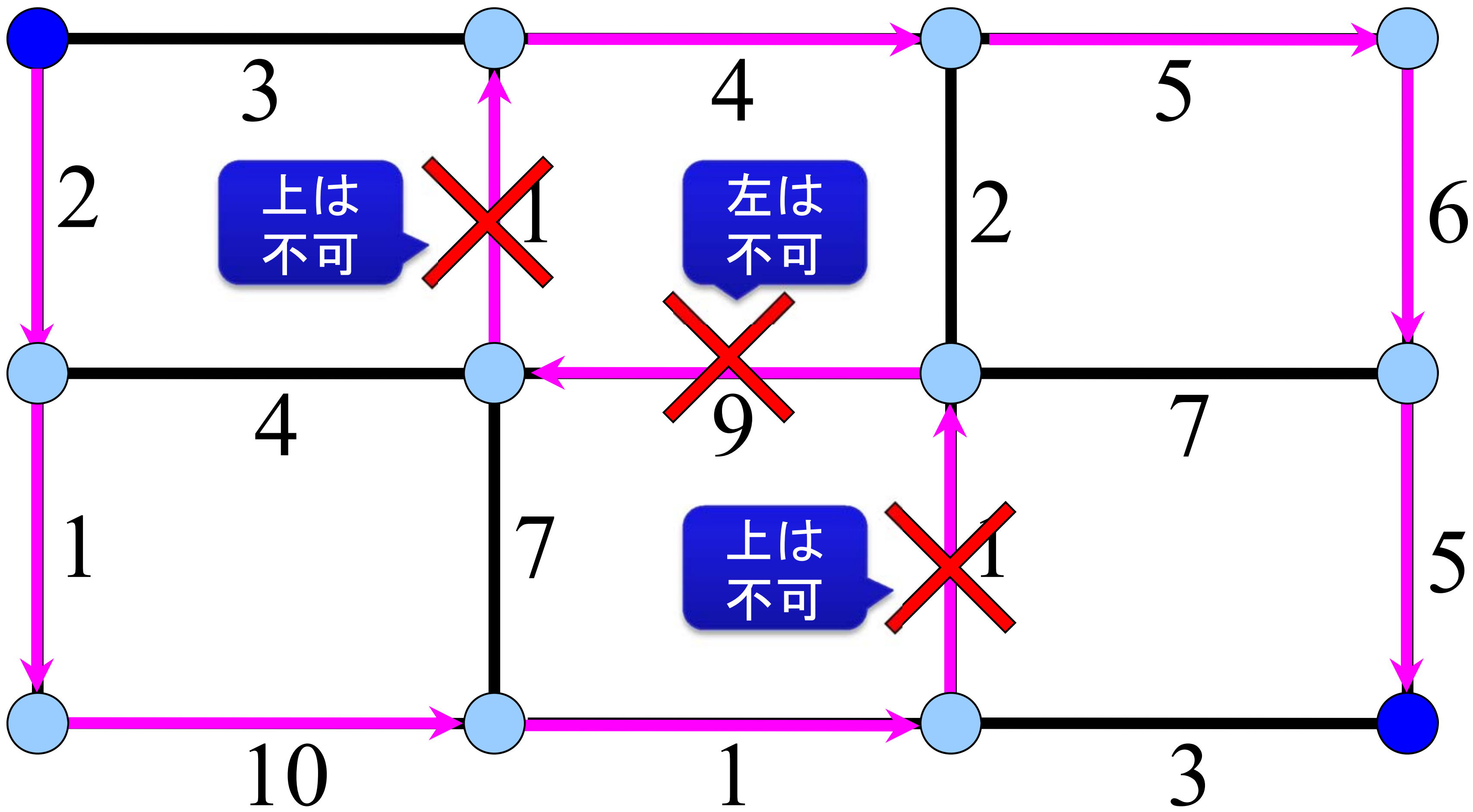
- 格子状のネットワーク
- 出発地：左上点，目的地：右下点
- 移動は右・下方向へのみ



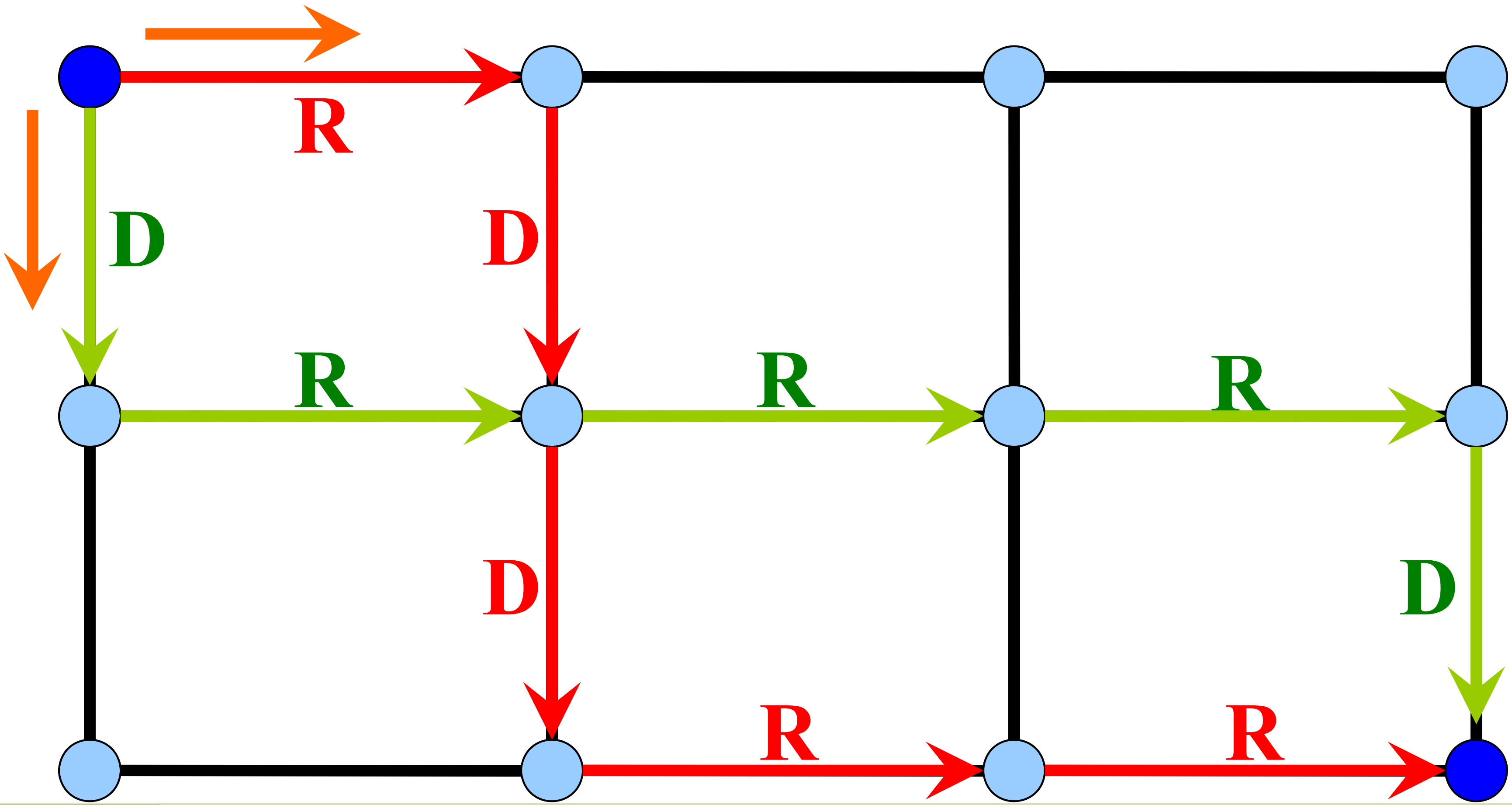
難しいなら易しくすればいいのさ！



難しいなら易しくすればいいのさ！



さて，経路は全部で幾つあるのか？



Point: どんな経路も，順番を無視すれば，R=3回，D=2回使う

緑の経路＝DRRRD

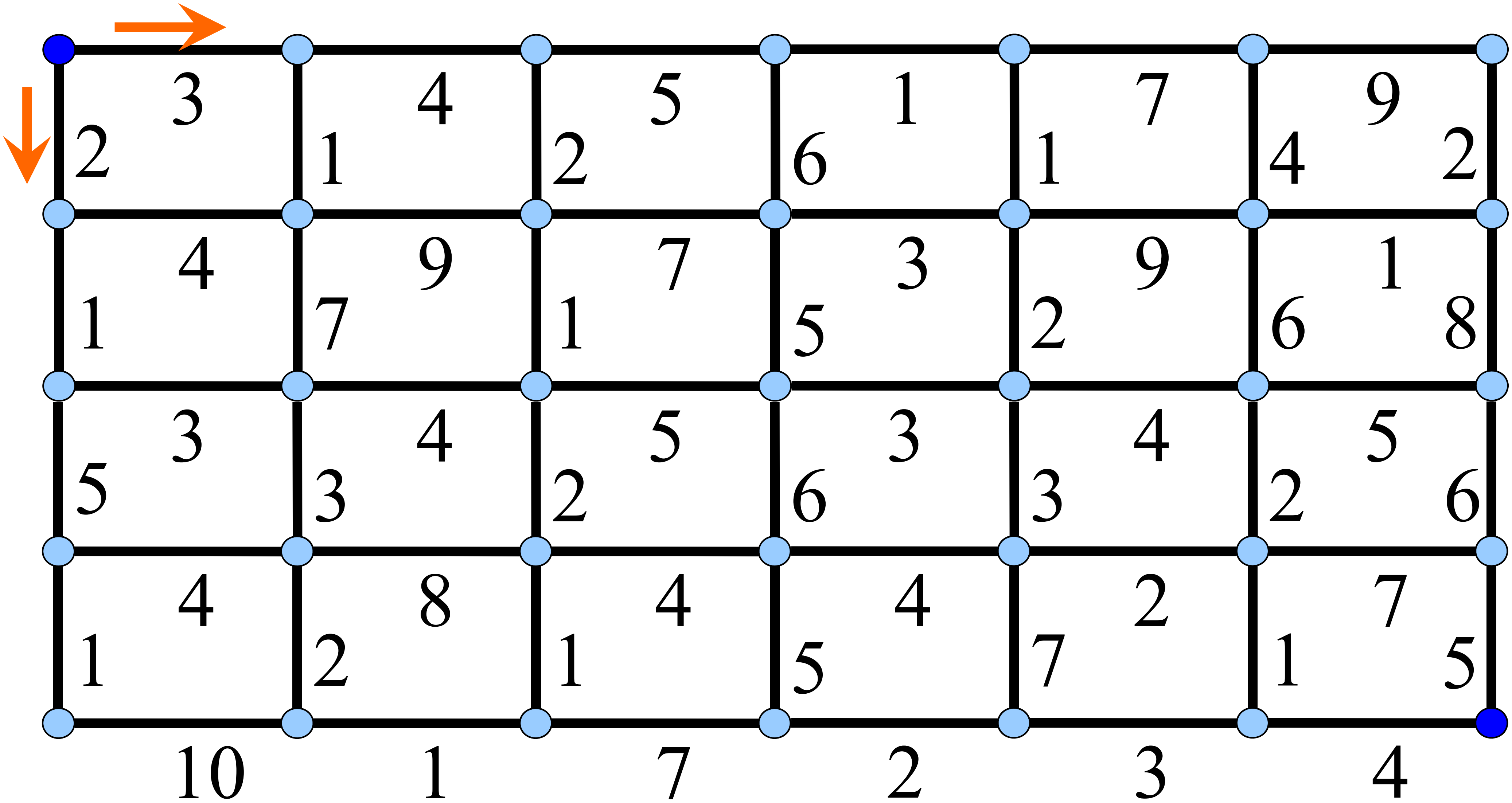
赤の経路＝RDDRR

i.e., (R+D)箇所のうちD箇所の置く場所を決めれば良い→ ${}_{R+D}C_D$

$$= \frac{(R+D)!}{R!\,D!}$$ 通り

例では ${}_{3+2}C_2 = 10$ 通り

経路は全部で幾つ？

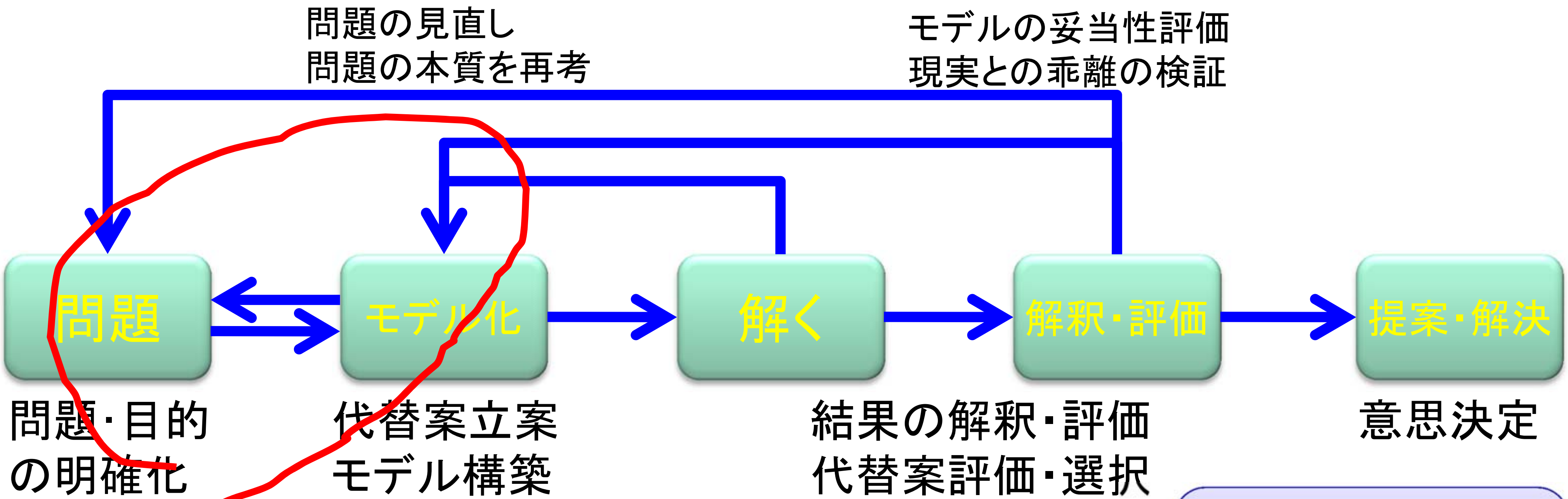


R=6, D=4なので，${}_{6+4}C_4$= 210 通り

意思決定

論理的思考力
データ分析，統計学
数理的アプローチ

- 「問題の把握」から「意思決定」までの流れ



- 問題発見・状況認識
  - 状況を把握し問題の背後にある本質を追究
  - いったい何を知りたいのか？
  - 問題の本質は何か？

現状認識力
問題発見・定義

- 推論・モデル作成
  - 推論に基づきモデル作成
  - 現実を支配する法則を数量的に明確化

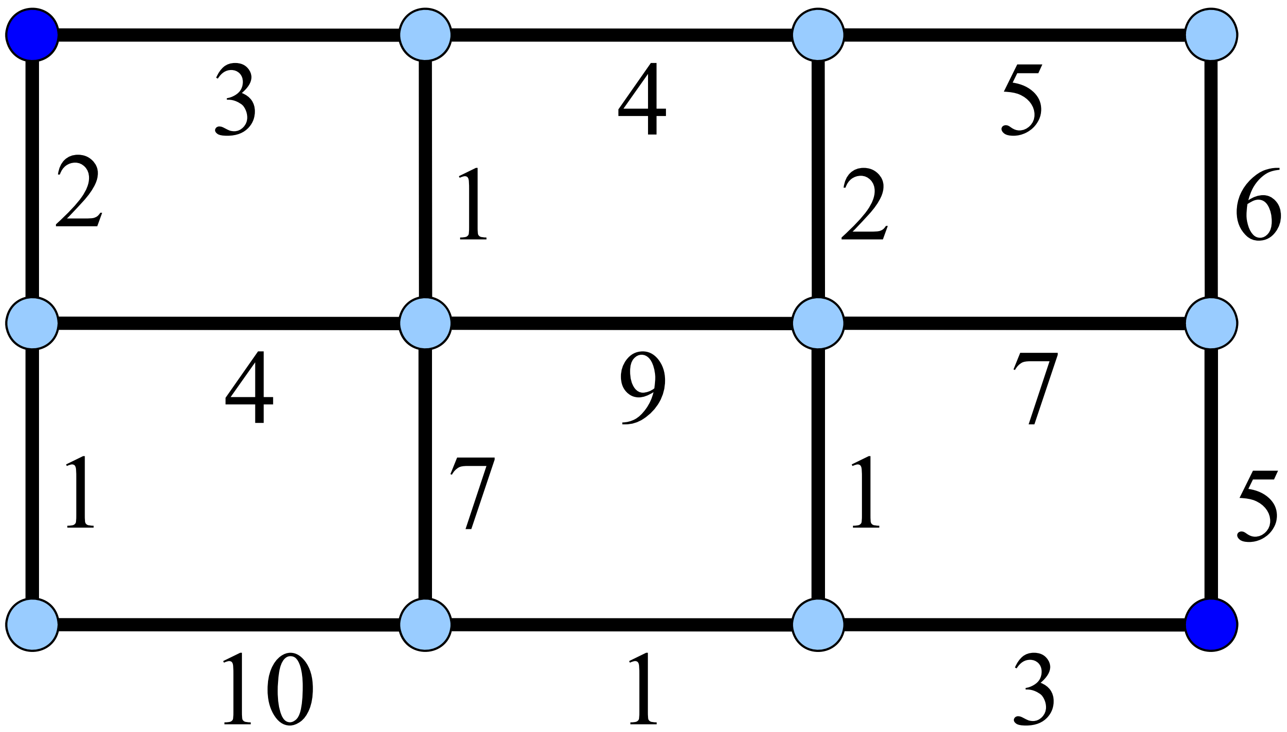
- 答えを導く
  - 解法選択
  - 解法構築
  - パラメータ調整

説得力
問題解決力

- 結果評価・解釈
  - 解法のもたらす結果の解釈・考察
  - 得られた代替案の評価・分析

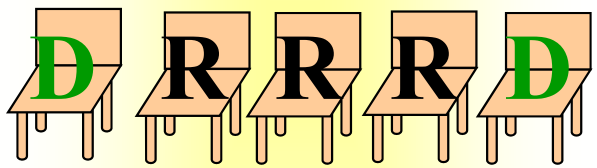
演習：やってみよう！

- **Q**: スタート（左上）からゴール（右下）へと至る最短経路を求めなさい．そしてそれが最短だと示しなさい



DRRRD ←これが1経路に対応

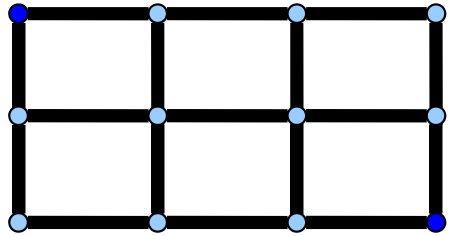
! 左の経路をなぞって探すのではなく，5脚の椅子にDを座らせる位値を決める



まずは，予め計算したとおり，<u>10通りのDの置き方</u>を全て書きだそう

経路は全部で幾つ？

R（横）	D（縦）	全経路
3	2	10
6	4	210
10	5	3,003
20	10	30,045,015
50	50	1.0E+29
100	100	9.1E+58
500	500	2.7E+299
1000	1000	#NUM!



【格子道路の街】
cf.京都市,札幌市
R, D幾つぐらい？

経路は全部で幾つ？

経路がとてもたくさんあるとは言っても，今のコンピュータは**かなりの速さで計算できる**んでしょ？　だから大丈夫だよね！

- 代表的なCPU, Game機, super computer の浮動小数点演算回数
  - Intel Core i7(3.2GHz) : **51.2GFLOPS** …1秒間に約**512億**回
  - PS3 : **218GFLOPS** …1秒間に約**2180億**回
  - PS4 : **1.84TFLOPS** …1秒間に約**1兆8400億**回
  - 京 : **10.51PFLOPS** …1秒間に約**1京510兆**回

（※2011年6月, 11月世界最速！ by Top500.org ）
（※2012年6月=2位，11 月=3位，2013年6月=4位，11月=4位）

〔Wikipedia「FLOPS」より〕
2013/5/1の情報

※**FLOPS = *FLoating-point Operations Per Second***

1つの経路を見つけ，その総コストを計算するのに，たどる経路枝数の浮動小数点演算でできると仮定しよう

例えば，R=10, D=5の経路なら，10+5回の演算で計算可と仮定するということ

K（キロ）≃ × 10^3=千倍
M（メガ）≃ × 10^6=百万倍
G（ギガ）≃ × 10^9=10億倍
T（テラ）≃ × 10^{12}=1兆倍
P（ペタ）≃ × 10^{15}=千兆倍
E（エクサ）≃ × 10^{18}=百京倍

経路は全部で幾つ？

R（横）	D（縦）	全経路	PS4 1.84TFLOPS	京 10.51PFLOPS
3	2	10	0.000000000 秒	0.000000000 秒
6	4	210	0.000000001 秒	0.000000000 秒
10	5	3,003	0.000000024 秒	0.000000000 秒
20	10	30,045,015	0.000489864 秒	0.000000086 秒
25	25	1.3E+14	57.25 分	0.601382523 秒
30	30	1.2E+17	44.63 日	11.25 分
40	40	1.1E+23	148,218.75 年	25.95 年
50	50	1.0E+29	1.73872E+11 年	30,439,996 年
100	100	9.1E+58	2.3E+31 宙齢	4.0E+27 宙齢
500	500	2.7E+299	3.4E+272 宙齢	5.9E+268 宙齢

圧倒的な計算力をもつコンピュータですら，力業（しらみつぶし）では答えを求めることが出来ない！

＃1宙齢＝138億年

ではどうする？

- 素朴で素直な方法〔列挙法〕
  - 全経路をしらみつぶしに調べて，最も短い経路を見つける方法

時間が掛かり過ぎる！

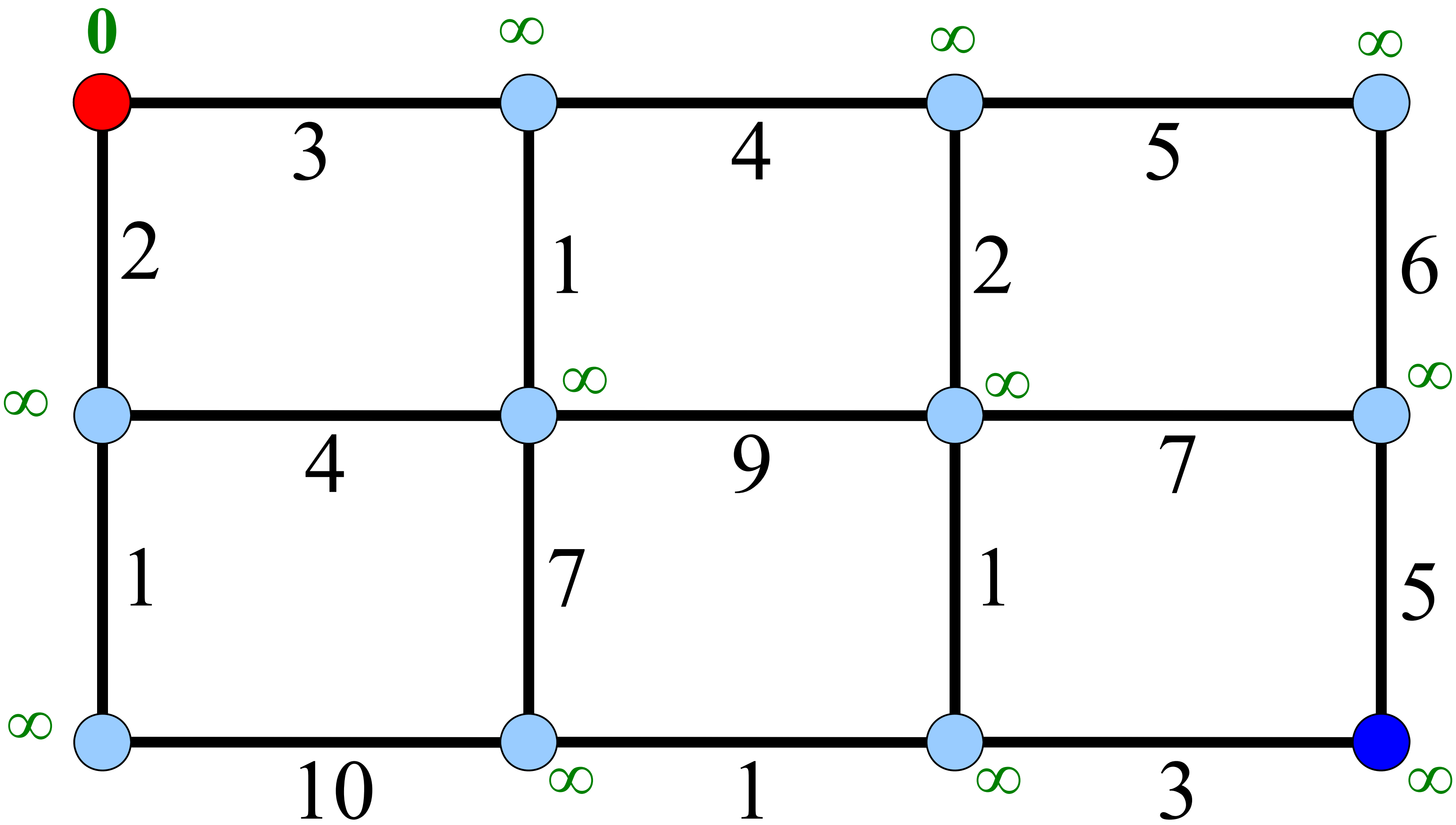
全経路をしらみつぶしに調べずに，最も短い経路を，現実的時間で見つける方法があるか？

Dijkstra法
（ダイクストラ法）

人間の創造的な仕事！

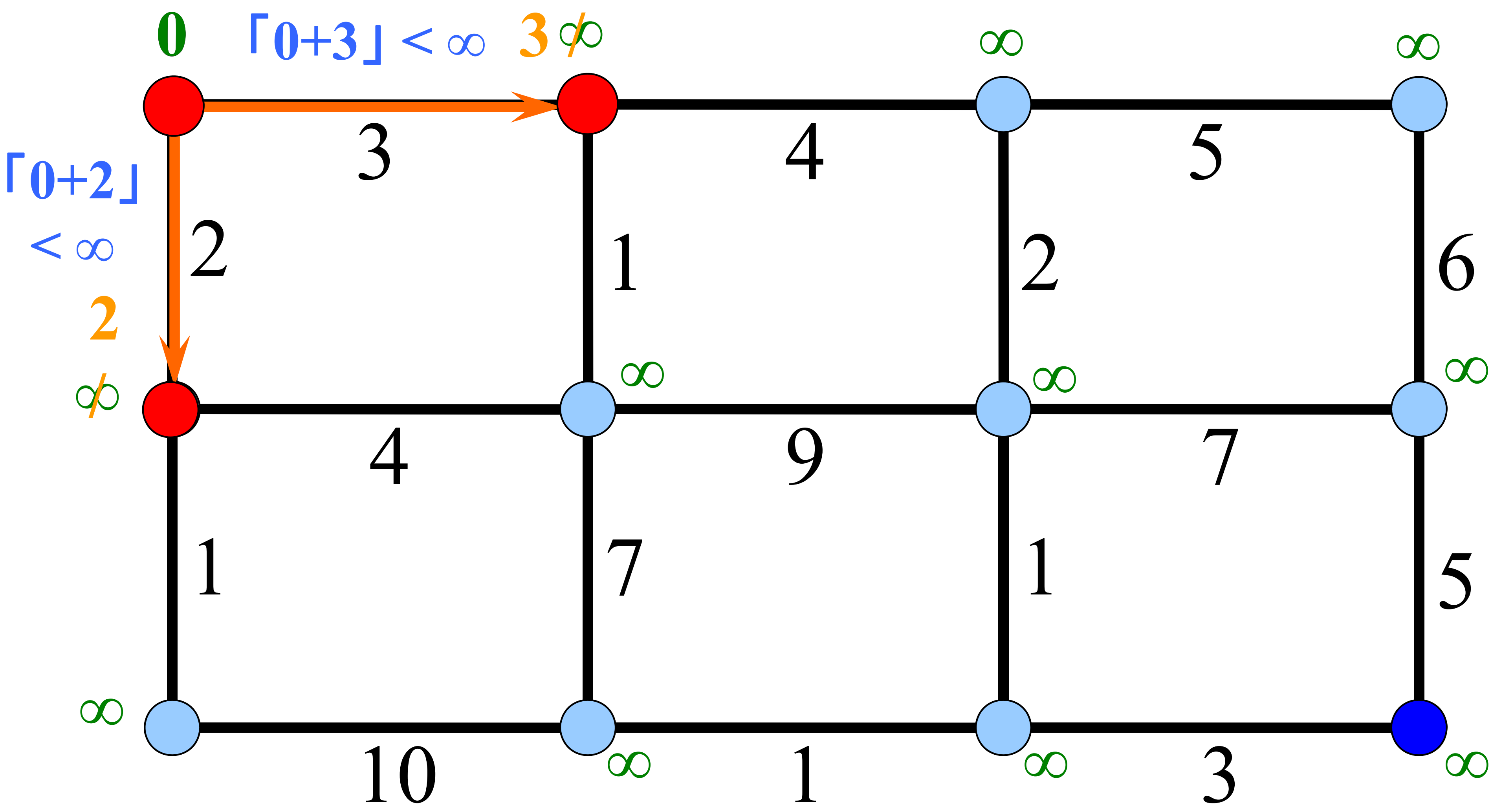
Dijkstra法
（初期設定）

step0: startのラベル＝0
その他のラベル＝∞
start点を調査中（●）に



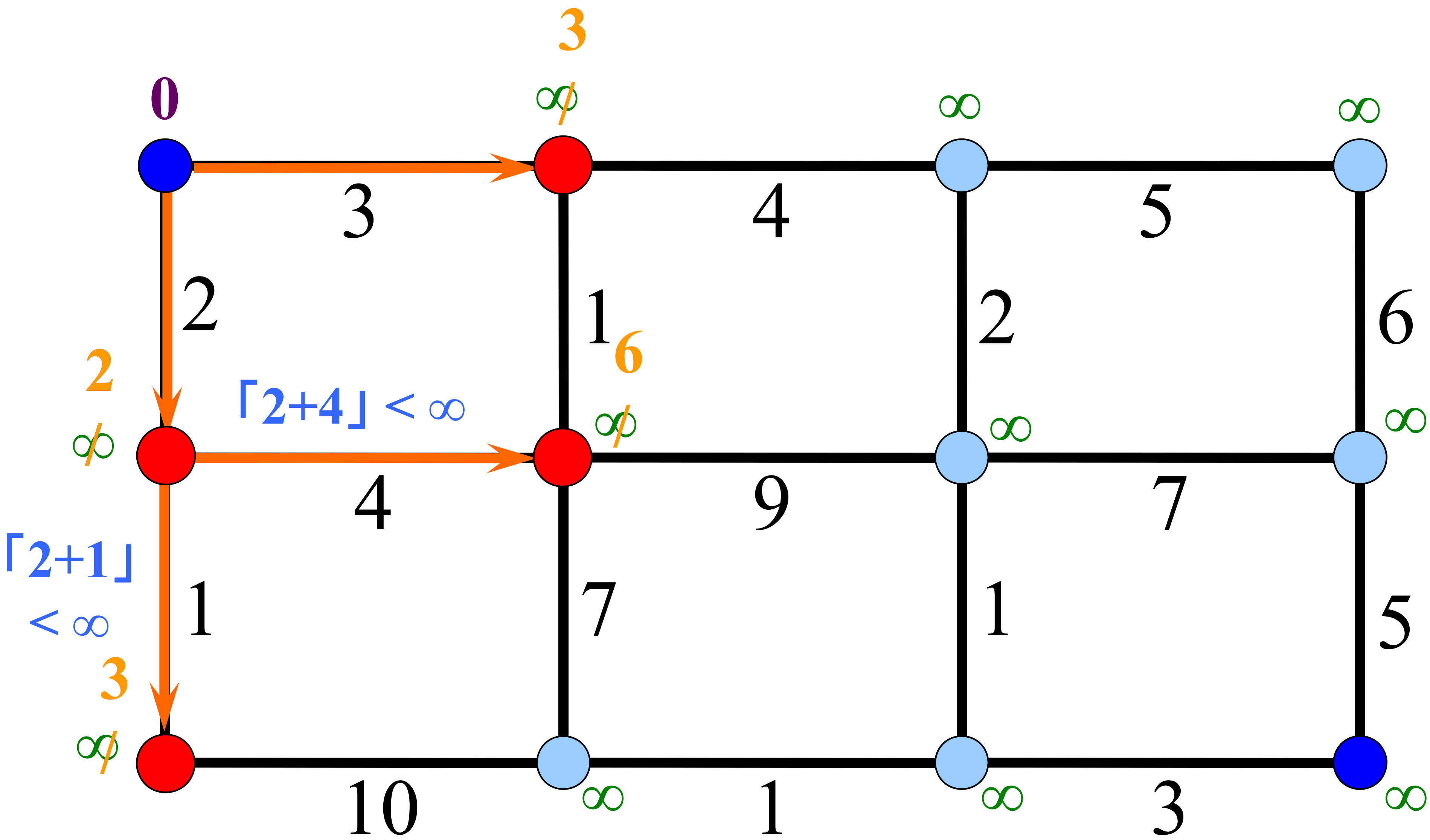
Dijkstra法（更新法）

step1-1: 調査中の点（ ）の中で，ラベルの値が最も小さい点を見つける
step1-2: その点から出る各枝について，「ラベル＋枝コスト」を計算し，枝先点のラベル値と比較，小さければ枝をオレンジにしてラベル更新，値を更新した点は調査中（ ）へ
step1-3: 全枝終了後，調査中から外し確定（値は紫色へ）



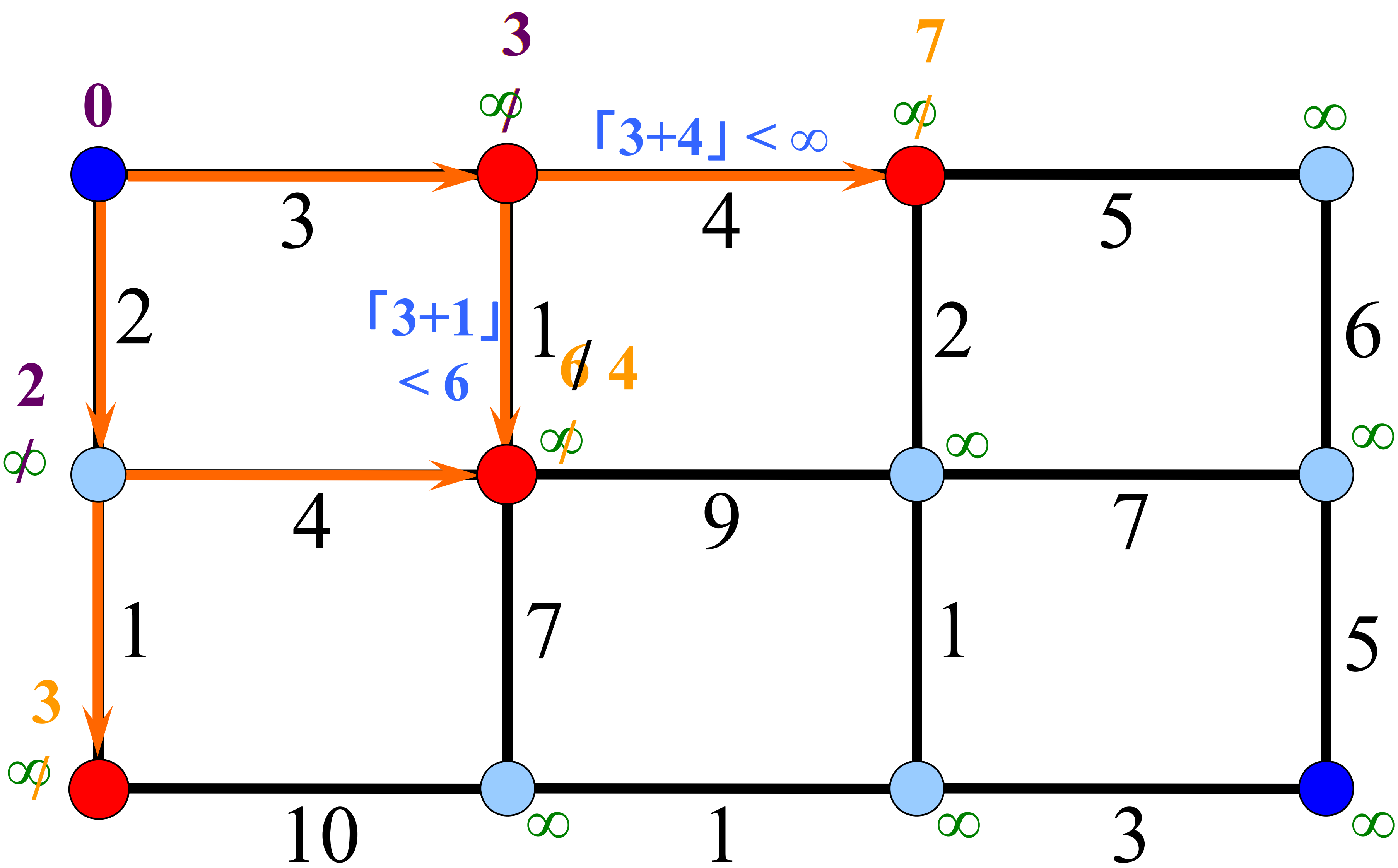
Dijkstra法

step1-1 〜 step1-3 を繰り返す



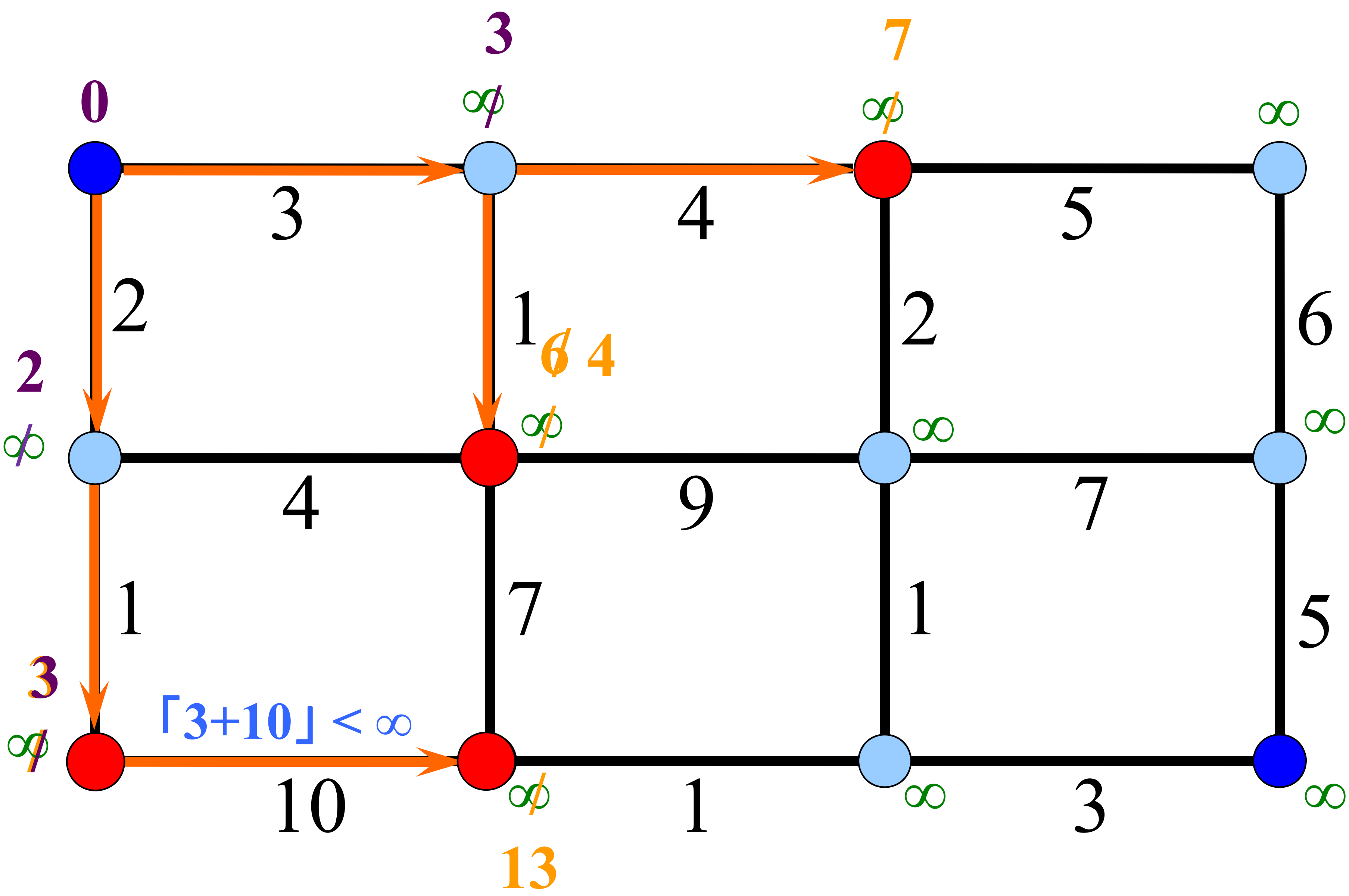
Dijkstra法

step1-1 ～ step1-3 を繰り返す



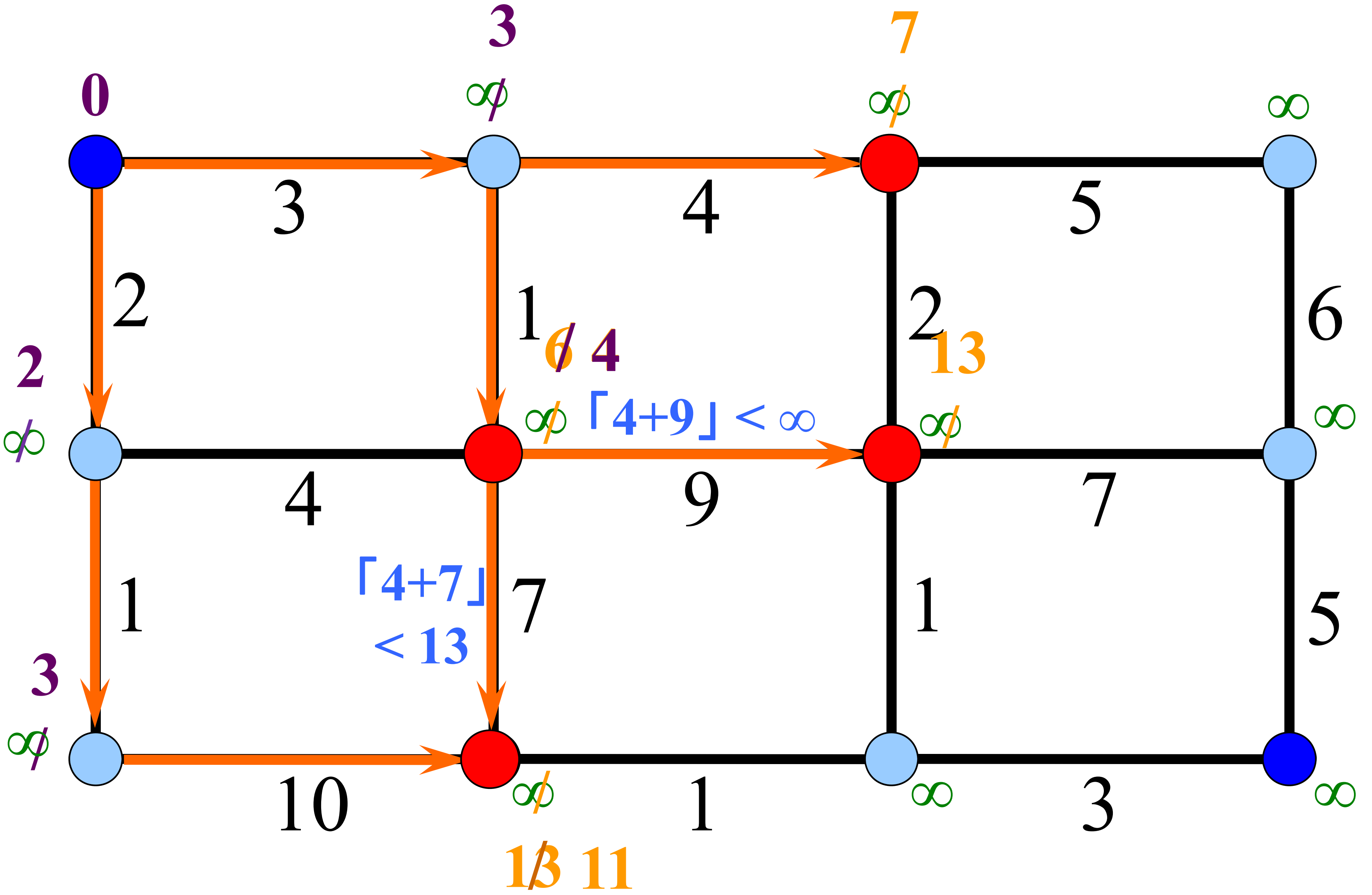
Dijkstra法

step1-1 ～ step1-3 を繰り返す



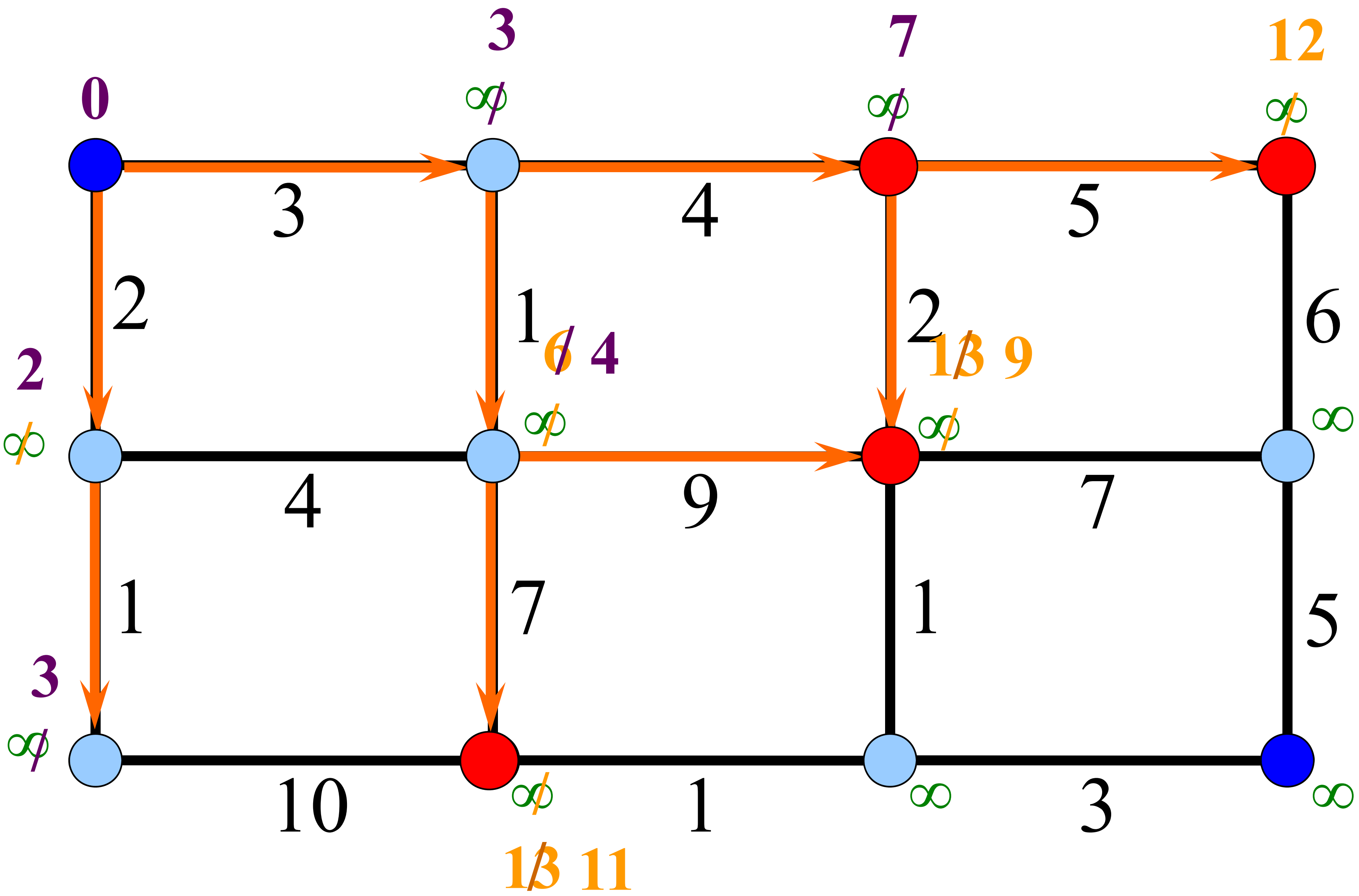
Dijkstra法

step1-1 ～ step1-3 を繰り返す



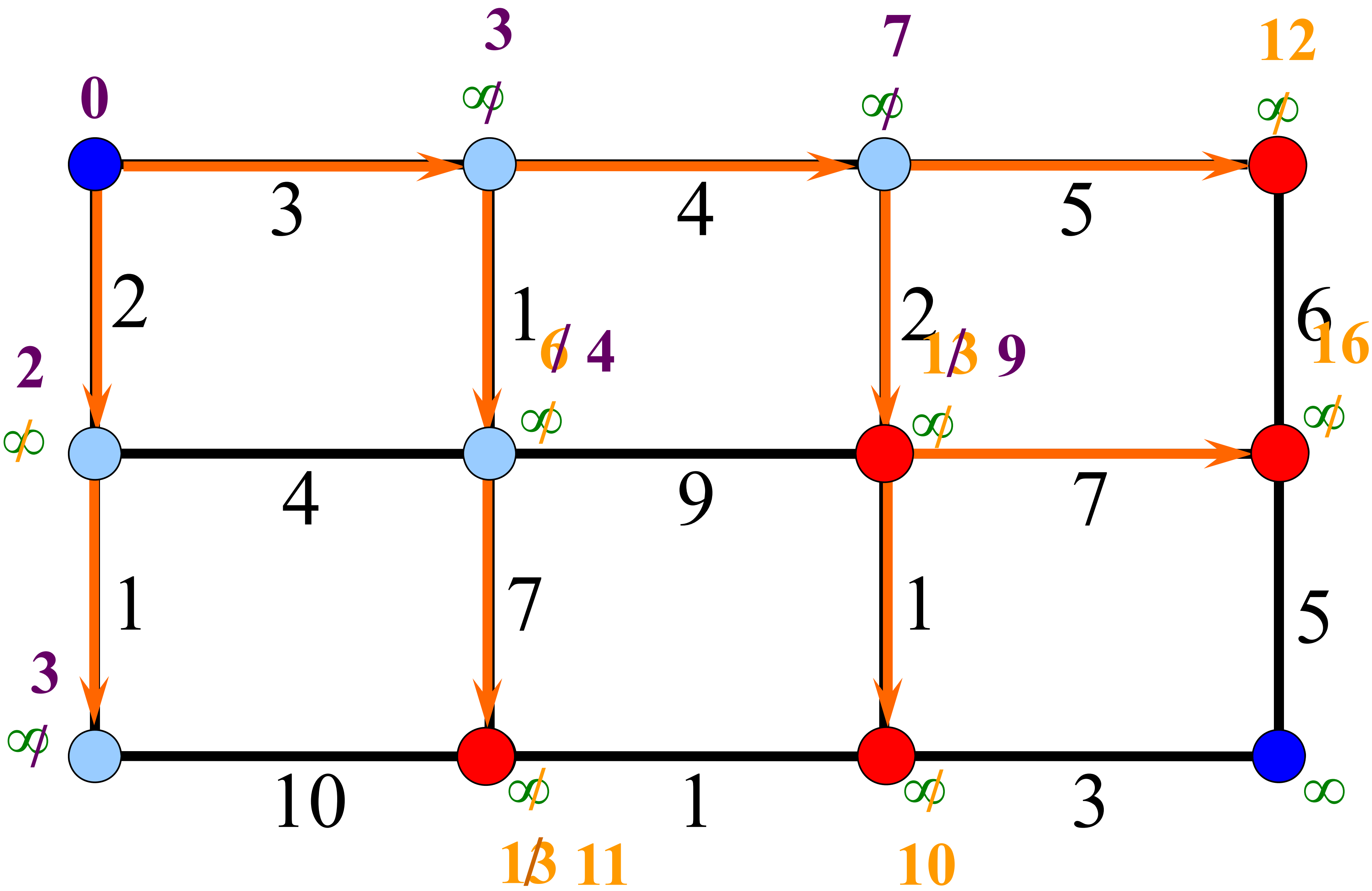
Dijkstra法

step1-1 ～ step1-3 を繰り返す



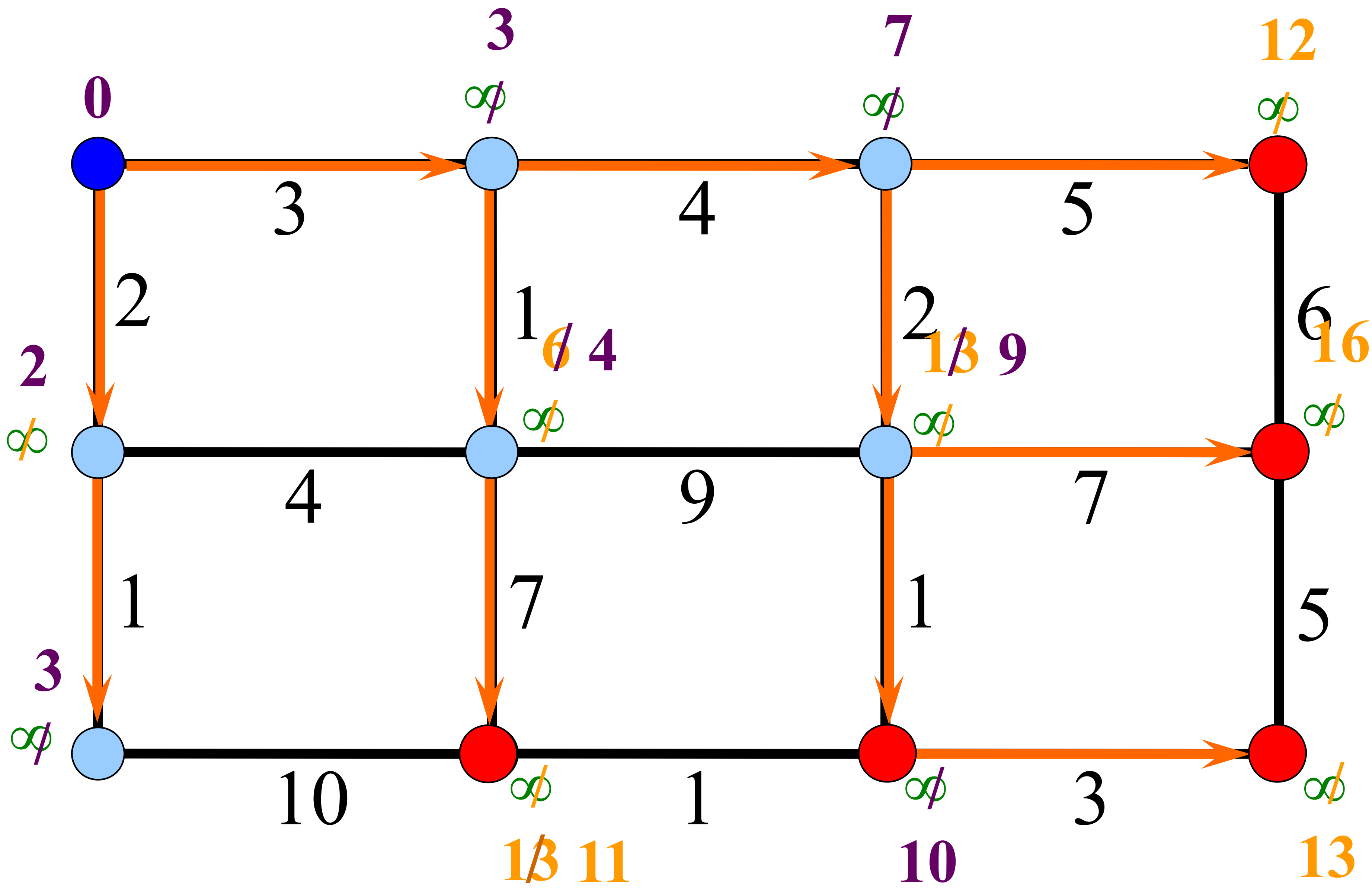
Dijkstra法

step1-1 ～ step1-3 を繰り返す



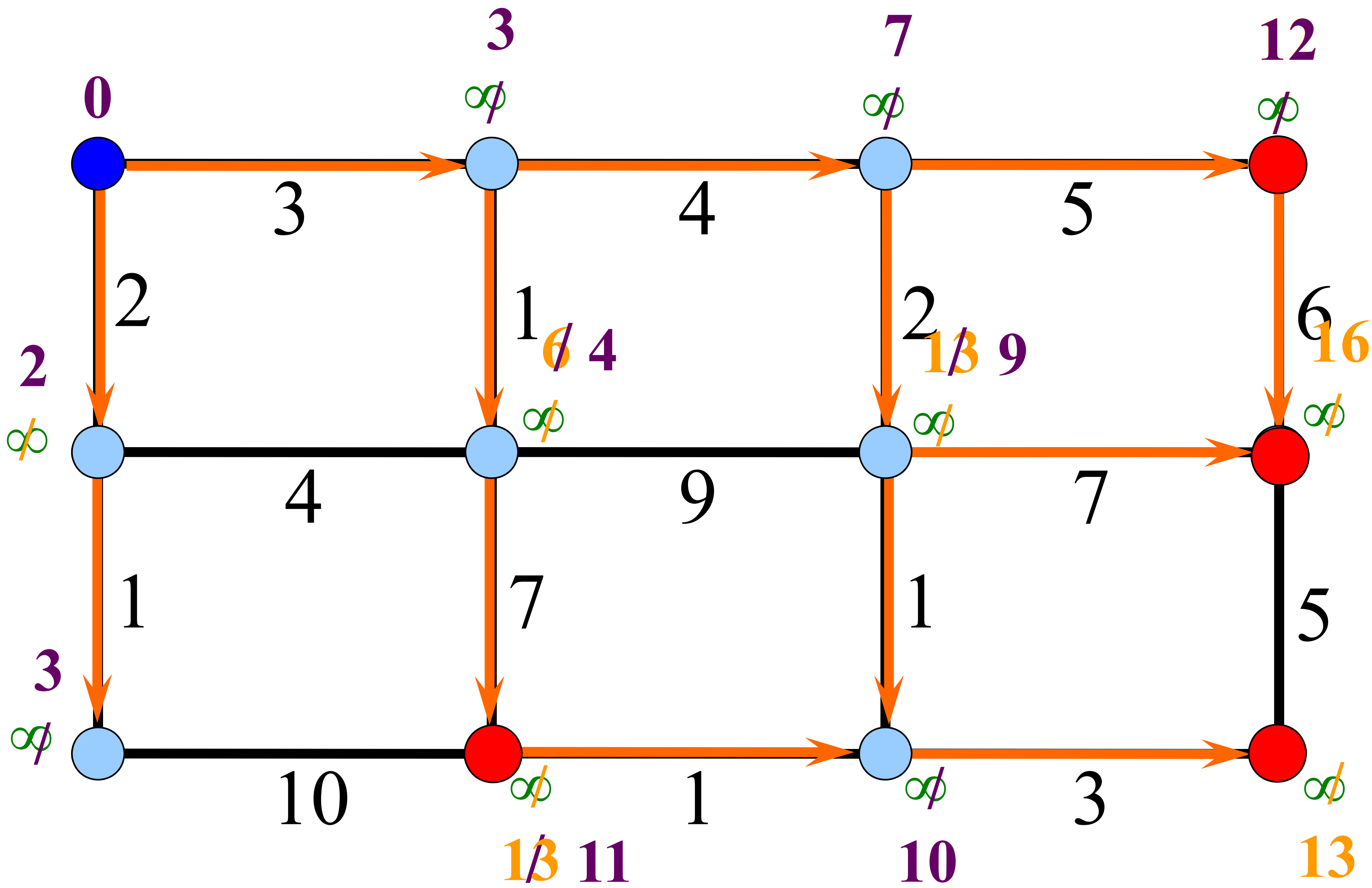
Dijkstra法

step1-1 ～ step1-3 を繰り返す



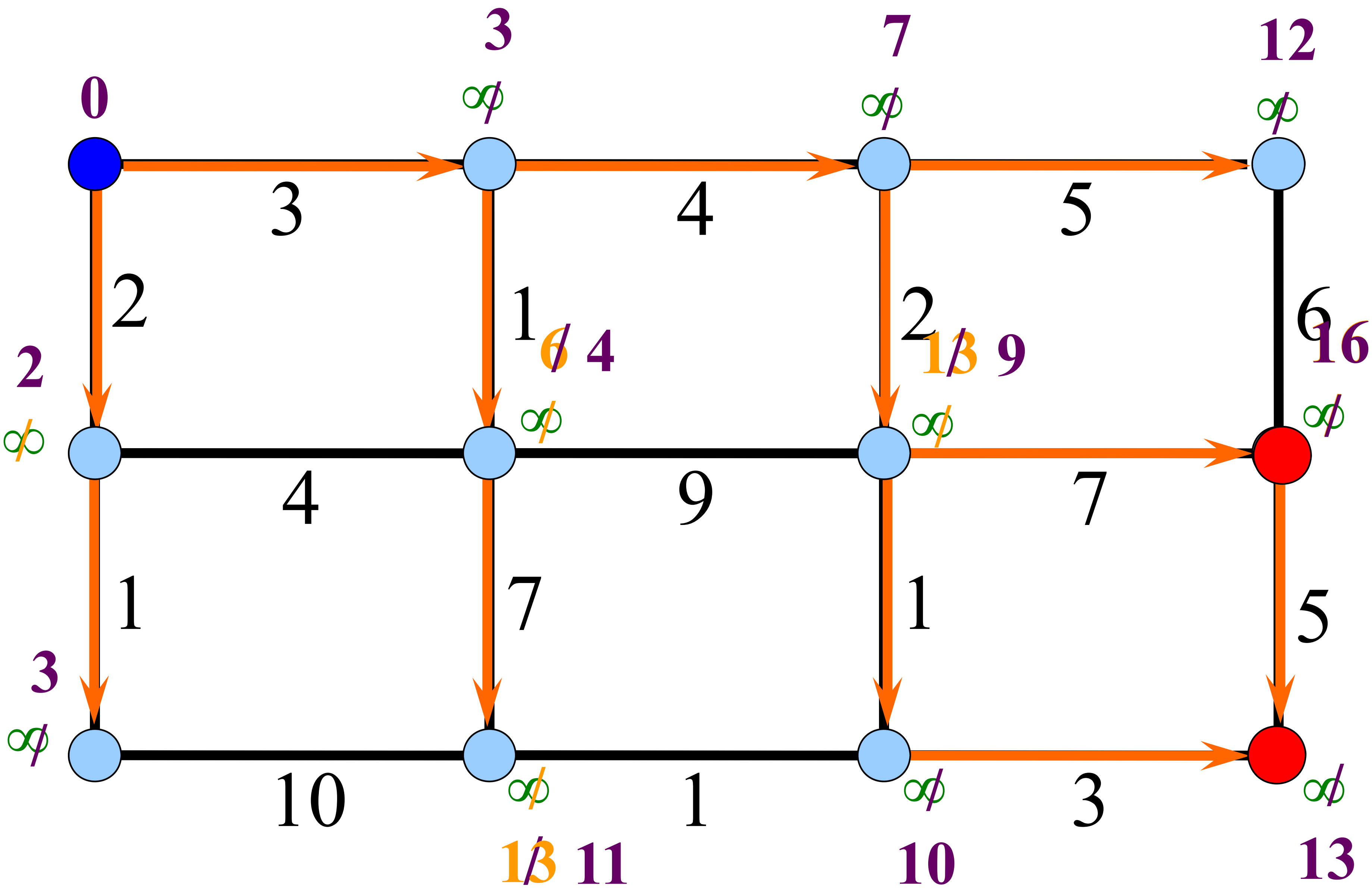
Dijkstra法

step1-1 ～ step1-3 を繰り返す



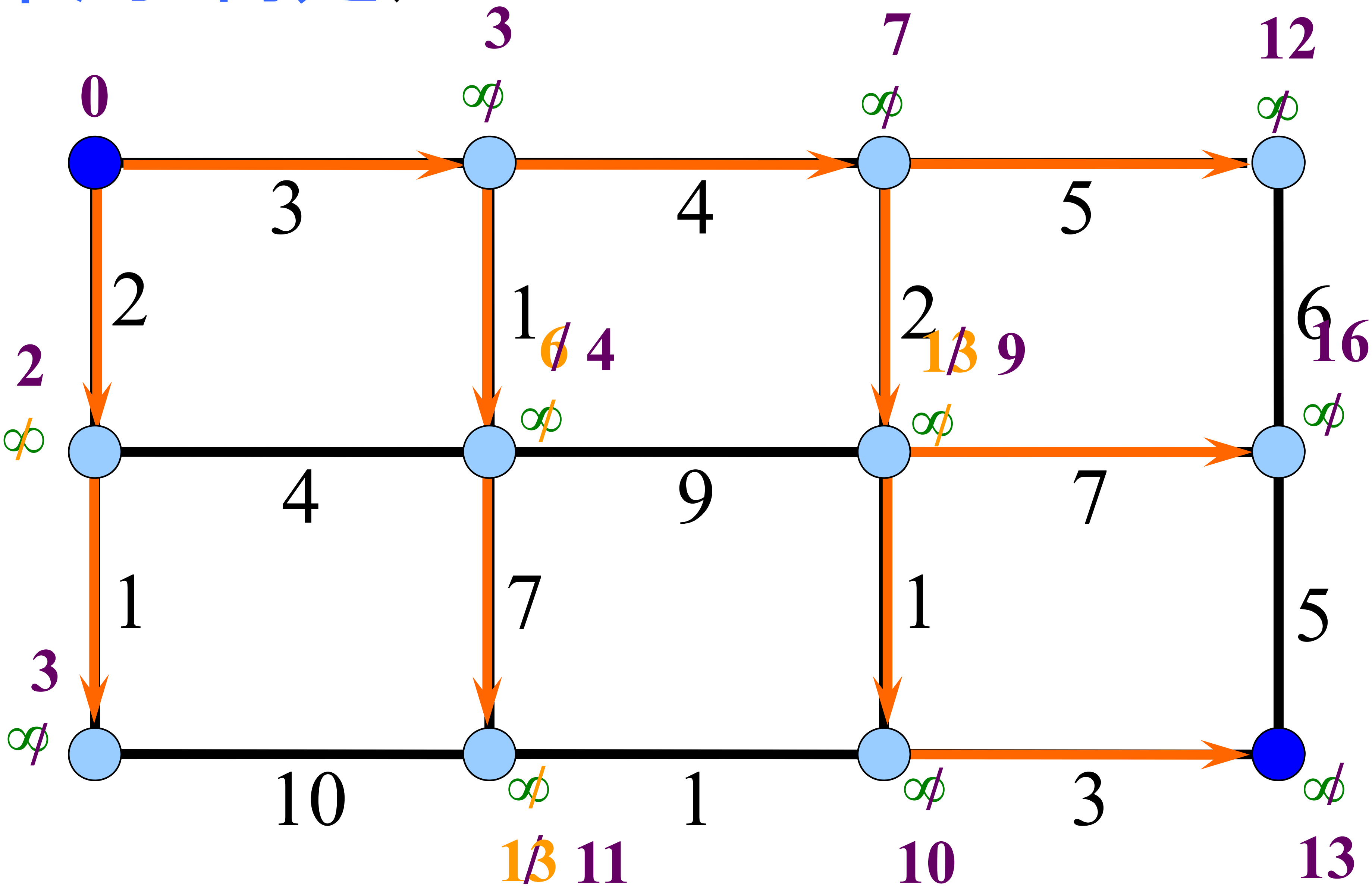
Dijkstra法

step1-1 ～ step1-3 を繰り返す



Dijkstra法（終了判定）

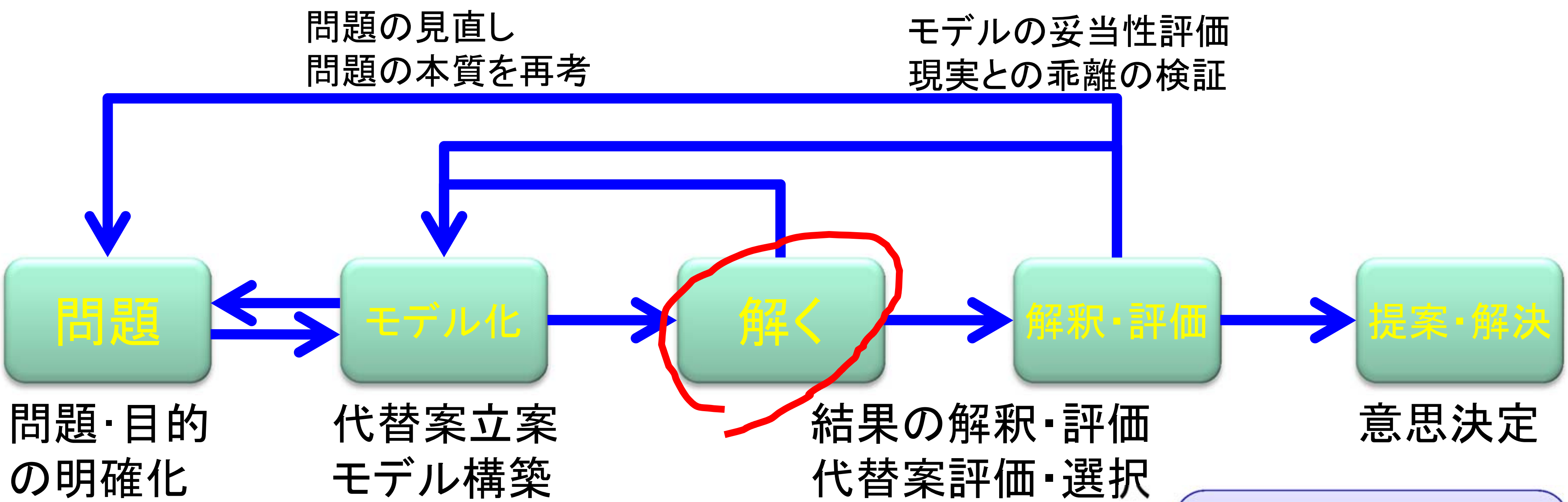
step2: 調査中の点（●）がなくなったら終了



意思決定

論理的思考力
データ分析，統計学
数理的アプローチ

- 「問題の把握」から「意思決定」までの流れ



- 問題発見・状況認識
  - 状況を把握し問題の背後にある本質を追究
  - いったい何を知りたいのか？
  - 問題の本質は何か？

- 答えを導く
  - 解法選択
  - 解法構築
  - パラメータ調整

説得力
問題解決力

- 結果評価・解釈
  - 解法のもたらす結果の解釈・考察
  - 得られた代替案の評価・分析

現状認識力
問題発見・定義

- 推論・モデル作成
  - 推論に基づきモデル作成
  - 現実を支配する法則を数量的に明確化

Dijkstra法って速いのか？

- 点の数を n とすると，大雑把な見積もりで，

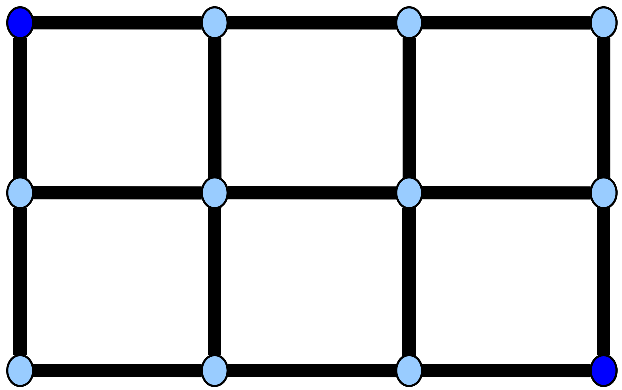
$$O(n^2)$$

多項式オーダー

$O(m+n\log n)$

- 点の数 n を右向枝数R，下向枝数Dで表すと

$$n=(R+1)\times(D+1)$$



$$n=(3+1)\times(2+1)=12$$

$$n^2=12^2=144$$

コンピュータに計算させてみよう！

簡単のため n^2 の5倍の浮動小数点演算回数で計算できると仮定.

Dijkstra法って速いのか？

R(横)	D(縦)	全経路	京 & しらみつぶし 10.51PFLOPS	Core i7 & Dijkstra 51.2GFLOPS
3	2	10	0.000000000 秒	0.000000001 秒
6	4	210	0.000000000 秒	0.000000003 秒
10	5	3,003	0.000000000 秒	0.000000006 秒
20	10	30,045,015	0.000000086 秒	0.000000023 秒
25	25	1.3E+14	0.601382523 秒	0.000000066 秒
30	30	1.2E+17	11.25 分	0.000000094 秒
40	40	1.1E+23	25.95 年	0.000000164 秒
50	50	1.0E+29	30,439,996 年	0.000000254 秒
100	100	9.1E+58	4.0E+27 宙齢	0.000000996 秒
500	500	2.7E+299	5.9E+268 宙齢	0.000024512 秒

世界最速SuperComp
＋力技（しょぼい方法）

<<<

そこらのPC
＋人間の知恵

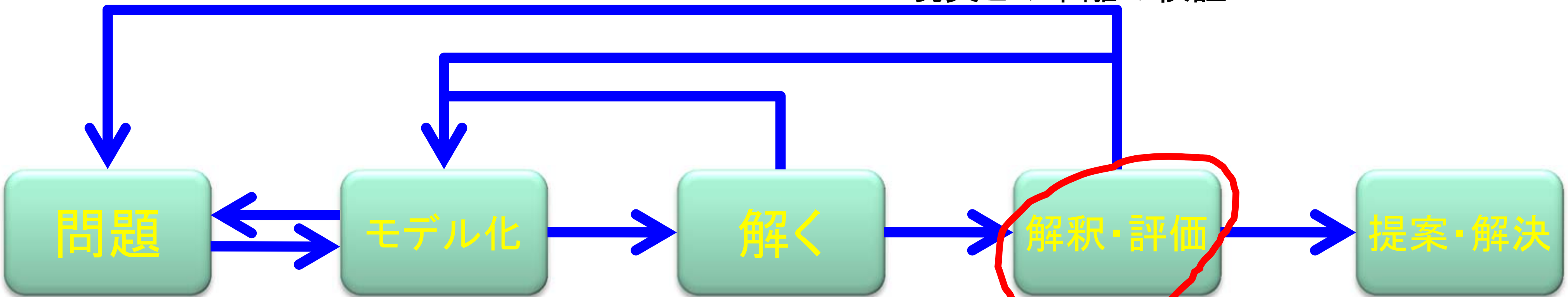
意思決定

論理的思考力
データ分析，統計学
数理的アプローチ

- 「問題の把握」から「意思決定」までの流れ

問題の見直し
問題の本質を再考

モデルの妥当性評価
現実との乖離の検証



問題・目的
の明確化

代替案立案
モデル構築

結果の解釈・評価
代替案評価・選択

意思決定

- 問題発見・状況認識
  - 状況を把握し問題の背後にある本質を追究
  - いったい何を知りたいのか？
  - 問題の本質は何か？

現状認識力
問題発見・定義

- 推論・モデル作成
  - 推論に基づきモデル作成
  - 現実を支配する法則を数量的に明確化

- 答えを導く
  - 解法選択
  - 解法構築
  - パラメータ調整

説得力
問題解決力

- 結果評価・解釈
  - 解法のもたらす結果の解釈・考察
  - 得られた代替案の評価・分析

意思決定支援･ビジネスサポート

カーナビは
存在しない

素朴な方法しか
ない世界

カーナビが
実現

Dijkstra法が
考案された
世界

人類の創造的
な仕事！

<u>参考文献</u>

コンピュータに仕事を奪われつつある人類…

[1] 新井紀子
「コンピュータが仕事を奪う」日経新聞社(2010)

[2] E. Brynjolfsson, A. McAfee, 村井章子訳
「機械との競争」日経BP社(2013)

もっと知りたい人へ

- 参考文献
  - グリッツマン，ブランデンベルク**「最短経路の本」** シュプリンガー（2008）
  - W.J.クック**「驚きの数学 巡回セールスマン問題」** 青土社（2013）
  - 久保，松井**「組合せ最適化『短編集』」** 朝倉書店（1999）
  - 山本，久保**「巡回セールスマン問題への招待」** 朝倉書店（1997）
  - 松井，根本，宇野**「入門オペレーションズ・リサーチ」**東海大出版（2008）

- 関連する授業
  - 「**ネットワークモデル分析**」（4セメ）
  - 「**最適化モデル分析**」（5セメ）　　　etc…