

関数型言語（OCaml 演習）(6)

ダイクストラ法

浅井 健一

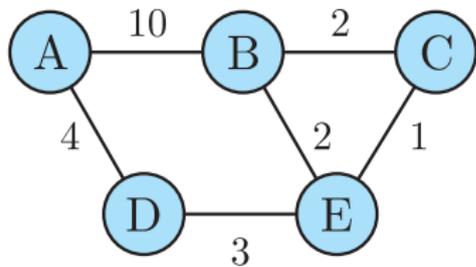
お茶の水女子大学

重みつきグラフと最短路

頂点 駅

辺 駅のつながり

重み その間の距離



最短路 始点から終点に至る道で、
通った辺の重みの和が最小のもの

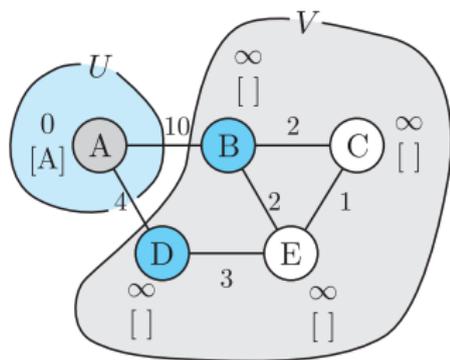
ダイクストラ法

入力 重み付きグラフと始点。

出力 始点から全点への最短距離と最短経路。

ダイクストラ法：準備

各頂点に最短距離と
手前リストの情報を付加。



1	最短距離	手前リスト
始点	0 (確定)	[始点]
その他	∞ (暫定)	[] (暫定)

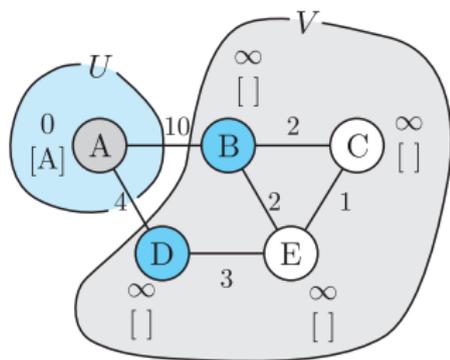
- 2 U: 最短距離が確定した点 (最初は始点のみ)
V: それ以外の点

アルゴリズムの概略

最短路の候補を、新しい道が見つかるたびに更新する

ダイクストラ法

直前に確定 A
更新 B ($\infty \rightarrow$
D ($\infty \rightarrow$
確定



メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を更新
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を確定し、 V から U に移す

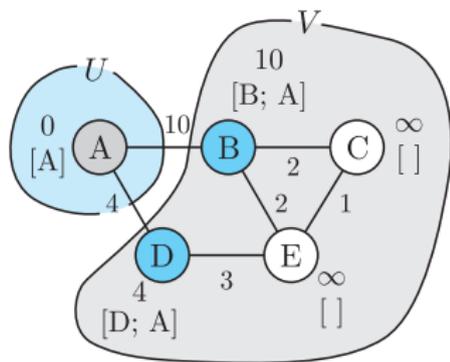
ダイクストラ法

直前に確定 A

更新 B ($\infty \rightarrow 10$),

D ($\infty \rightarrow 4$)

確定



メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を更新
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を確定し、 V から U に移す

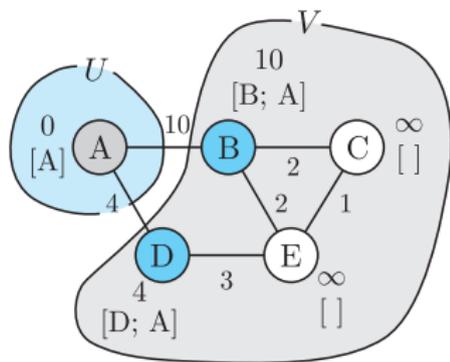
ダイクストラ法

直前に確定 A

更新 B ($\infty \rightarrow 10$),

D ($\infty \rightarrow 4$)

確定 D

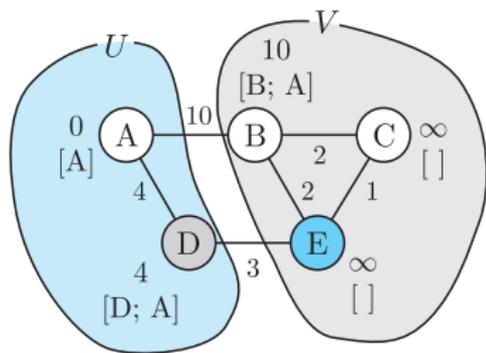


メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を更新
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を確定し、 V から U に移す

ダイクストラ法

直前に確定 D
更新 E ($\infty \rightarrow$
確定

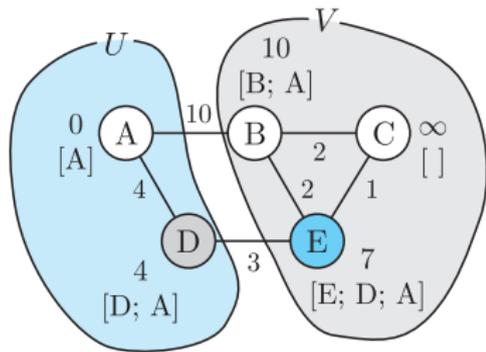


メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を**更新**
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を**確定**し、 V から U に移す

ダイクストラ法

直前に確定 D
更新 E ($\infty \rightarrow 7$)
確定

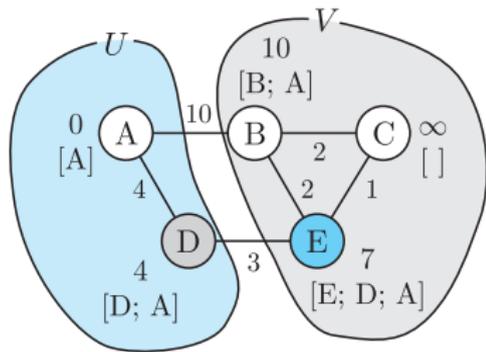


メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を更新
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を確定し、 V から U に移す

ダイクストラ法

直前に確定 D
更新 E ($\infty \rightarrow 7$)
確定 E



メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を**更新**
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を**確定**し、 V から U に移す

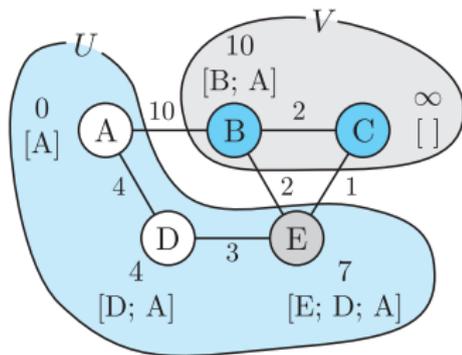
ダイクストラ法

直前に確定 E

更新 B (10 →

C (∞ →

確定



メインループ

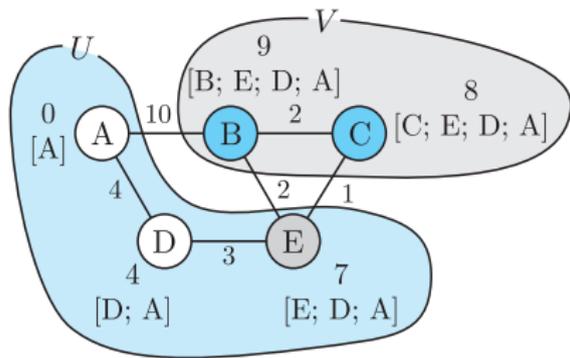
- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を更新
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を確定し、 V から U に移す

ダイクストラ法

直前に確定 E

更新 B (10 → 9),
C (∞ → 8)

確定



メインループ

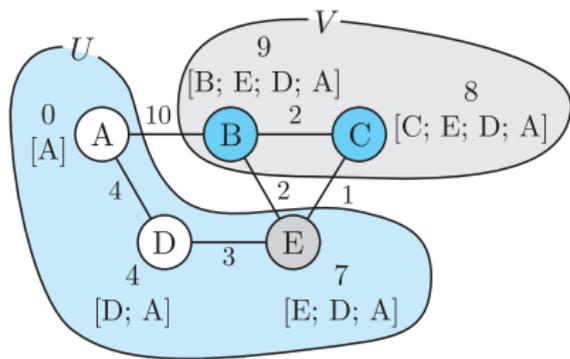
- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を更新
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を確定し、 V から U に移す

ダイクストラ法

直前に確定 E

更新 B (10 → 9),
C (∞ → 8)

確定 C

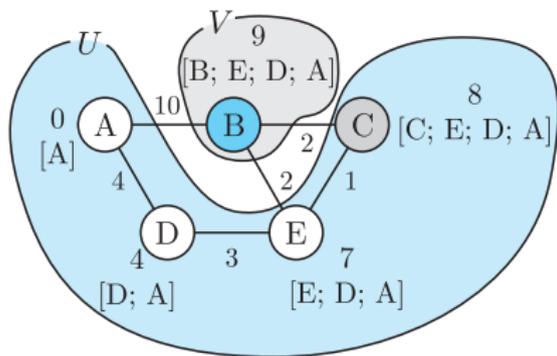


メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を**更新**
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を**確定**し、 V から U に移す

ダイクストラ法

直前に確定 C
更新 B (9 →
確定

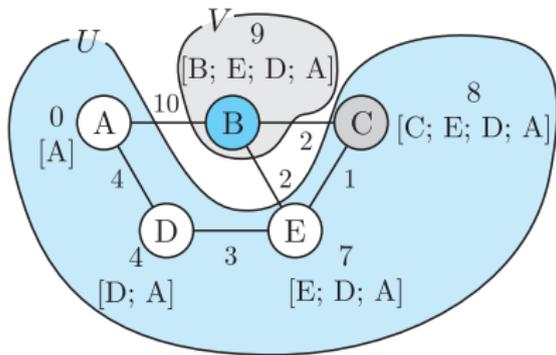


メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を更新
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を確定し、 V から U に移す

ダイクストラ法

直前に確定 C
更新 B (9 → 9)
確定

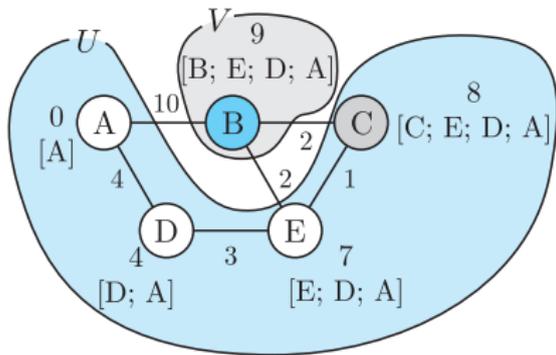


メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を更新
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を確定し、 V から U に移す

ダイクストラ法

直前に確定 C
更新 B (9 → 9)
確定 B

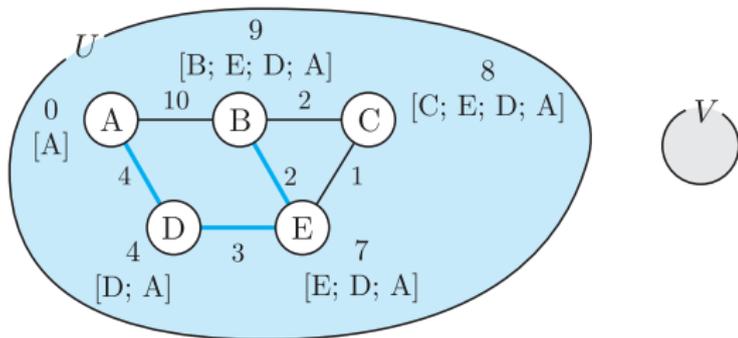


メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を更新
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を確定し、 V から U に移す

ダイクストラ法

V が空になったので終了。
始点 A から各点までの最短距離と最短経路が求まった。



メインループ

- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を**更新**
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を**確定**し、 V から U に移す

データ構造と必要な関数

頂点 駅名（文字列）、最短距離（実数）、手前リスト（駅名のリスト）を持つレコード

頂点の集合 頂点のリスト

辺 駅名（文字列）ふたつと間の距離（実数）を持つレコード

辺の集合 辺のリスト

U, V は頂点のリストで表現。必要な主な関数：

- 頂点の情報を初期化する関数
- 頂点の情報を更新する関数
- 最短距離最小の頂点を分離する関数

まとめ

ダイクストラ法

- 1 各頂点の最短距離、手前リストを初期化
- 2 集合 U (始点のみ) と V (それ以外の点) を用意
- 3 以下を未確定の点 V がなくなるまで繰り返す
- 4 直前に確定した点に接続している点の情報を**更新**
- 5 V の中で「最短距離最小の点 p 」を選択
- 6 p の情報を**確定**し、 V から U に移す

(点 p を V から U に移す時点で、 p の持っている最短距離が本当の最短距離になっていることを示せる。)