コンピュータ・サイエンス2

第14回
プログラミング・アルゴリズム(実習)(続き),
データモデル

人間科学科コミュニケーション専攻
白銀 純子

第14回の内容

- プログラミング・アルゴリズムの実習

設問1

⌘「逐次探索」とは何かを説明しなさい。

解答:
データを前から順に1つずつ比較して探していく方法

プログラミング実習

実習内容

- プログラミング実習
  - 命令書(プログラム)とはどのようなものか?
  - どうやって翻訳・通訳するか?
  - どうやって実行するか?
- アルゴリズム実習
  - アルゴリズムによって、そんなに処理時間に違いがあるか?

準備

- 授業のページから4つのファイルをダウンロード
  - BubbleSort.c
    - バブルソートをするC言語のプログラム
  - BubbleSort.html
    - バブルソートをするJavaScriptのプログラム
  - MergeSort.c
    - 併合ソートをするC言語のプログラム
  - MergeSort.html
    - 併合ソートをするJavaScriptのプログラム

※リンクをクリックするのではなく、右クリック→「ファイルをダウンロード」でダウンロードすること

C言語とJavaScript

- C言語
  - プログラミング言語の一種
  - 記述された命令書を機械語に翻訳した命令書を作成する形式の言語(コンパイラ型の言語)
- JavaScript
  - プログラミング言語の一種
  - 記述された命令書を機械語に通訳する形式の言語(インタプリタ型の言語)
  - Webページでよく利用

ファイルの役割

- BubbleSort.c
  - 「BubbleRandomNum.txt」に、並び替え前の数を保存
  - 「BubbleSortNum.txt」に、並び替え後の数を保存
- BubbleSort.html
  - 並び替え前と後の数をブラウザに表示
- MergeSort.c
  - 「MergeRandomNum.txt」に、並び替え前の数を保存
  - 「MergeSortNum.txt」に、並び替え後の数を保存
- MergeSort.html
  - 並び替え前と後の数をブラウザに表示

BubbleSort.c, MergeSort.c[1]

- コンパイルして機械語のファイルを作成
  1. Finder→「アプリケーション」→「ユーティリティ」→「ターミナル」をダブルクリック
  2. それぞれのファイルをコンパイル
     - BubbleSort.cの場合:
       gcc -o BubbleSort BubbleSort.c
       と入力し、「Return」キーを押す
     - MergeSort.cの場合:
       gcc -o MergeSort MergeSort.c
       と入力し、「Return」キーを押す

BubbleSort.c, MergeSort.c[2]

3. Finderで、「BubbleSort」ファイルと「MergeSort」ファイルができていることを確認
   - ⌘BubbleSort: BubbleSort.cを機械語に翻訳したファイル
   - ⌘MergeSort: MergeSort.cを機械語に翻訳したファイル

BubbleSort.c, MergeSort.c[2]

- プログラミング言語が機械語に翻訳されているかを確認
  - BubbleSort.cをJeditで開いてみる
  - BubbleSortをJeditで開いてみる
  - MergeSort.cをJeditで開いてみる
  - MergeSortをJeditで開いてみる

- gcc: C言語のプログラムを機械語に翻訳するためのコンパイラ
  - ✓「gcc -o ファイル1 ファイル2」で、「ファイル2のプログラムを機械語に翻訳し、ファイルに保存する」という命令

BubbleSort.c, MergeSort.c[3]

- プログラムを実行
  - BubbleSort.cの場合: ターミナルで「./BubbleSort」と入力し、「Return」キーを押す
  - MergeSort.cの場合: ターミナルで「./MergeSort」と入力し、「Return」キーを押す

- BubbleRandomNum.txtとBubbleSortNum.txtを開き、数が並び替えされているかどうかを確認
- MergeRandomNum.txtとMergeSortNum.txtを開き、数が並び替えされているかどうかを確認
- 「Time: かかった時間 second」とターミナル上に表示されるので、かかった時間を比較

※かかった時間の単位は秒

BubbleSort.html, MergeSort.html

⌘BubbleSort.htmlとMergeSort.htmlをWebブラウザで表示

- ブラウザで表示された数が、「並び替え前」と「並び替え後」で並び替えられているかどうかを確認
- 「かかった時間msec」(ブラウザの一番下に表示)ので、BubbleSort.htmlとMergeSort.htmlでかかった時間を比較
- BubbleSort.cとBubbleSort.htmlでかかった時間を比較
  (※BubbleSort.cは数が30000個、BubbleSort.htmlは数が5000個)
- MergeSort.cとMergeSort.htmlでかかった時間を比較
  (※MergeSort.cは数が30000個、MergeSort.htmlは数が5000個)

Question!

データモデル(p. 142)

- モデル: 対象のある側面を模倣したもの
  - Ex. プラモデル: 何かの形をプラスチックで模倣したもの
- データモデル: データを単純化したモデル
  - 利用者が扱うデータの構成を単純化して表したもの
  - コンピュータの専門家以外でも利用
- 参考-データ構造: 処理の効率化のためのデータの構造
  - コンピュータ内部で扱うもの
  - コンピュータの専門家以外が利用することはあまりなし

データモデルの必要性(p. 145)

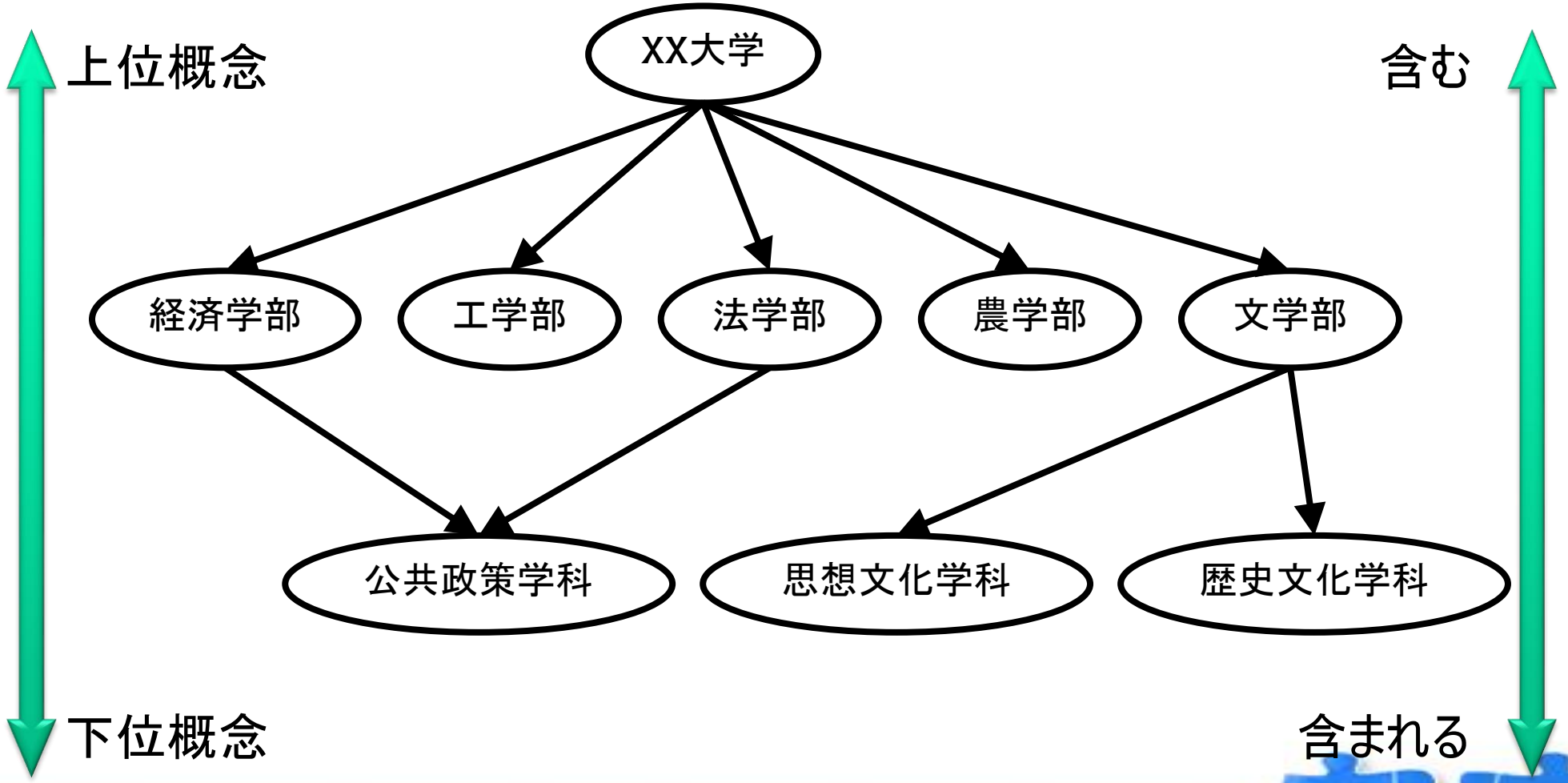
- データはコンピュータで処理
  - 専門家がプログラムを作成する必要
- データについて詳しいのは利用者
  - 利用者がデータモデルを理解できる必要
  - 利用者がデータの内容を専門家に伝える必要
  - 個人のデータをどのようなデータモデルで整理するかを考え、アプリケーションで処理する必要
    - 1つのデータを複数種類のデータモデルで表現することも可能

よく使われるデータモデル(p. 145)

- 階層モデル
- ネットワークモデル
- 関係モデル
- etc.

階層モデル[1](p. 146)

- 要素間の上下関係などの階層性を表したモデル
  - 楕円で表した1つ1つの要素: ノード

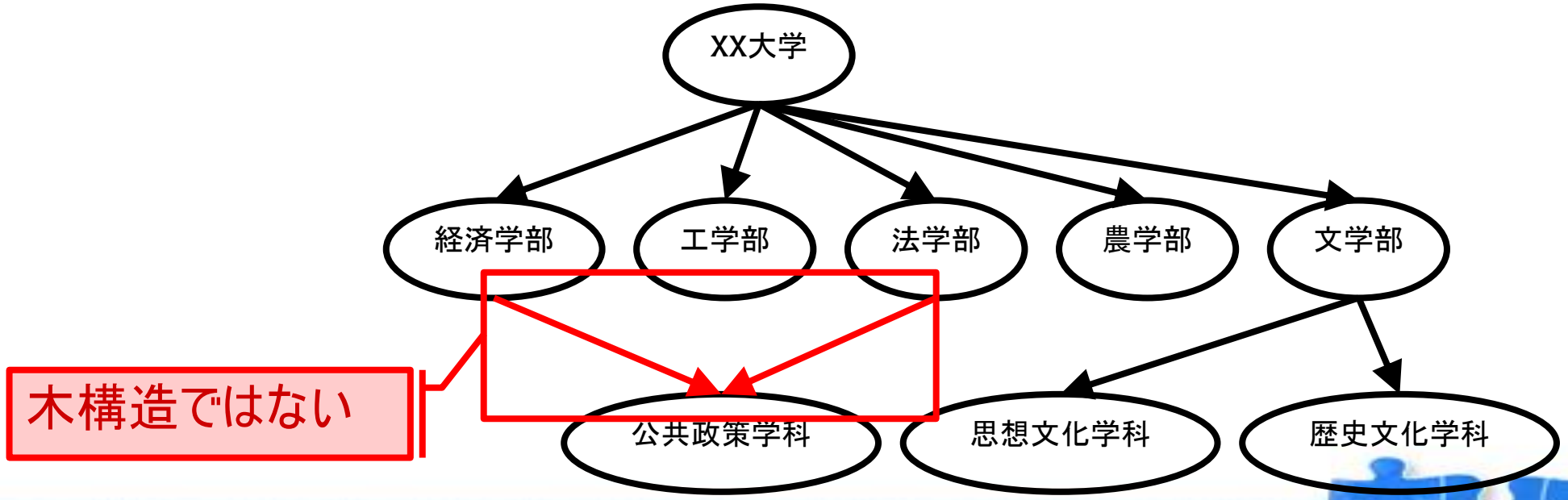


階層モデル[2](p. 146)

- 階層モデルでは…
  - データの包含関係や上下関係などを表現可能
    - Ex1. XX大学には経済学部、工学部、法学部…がある
    - Ex2. 文学部には、思想文化学科、歴史文化学科…がある
  - 包含関係・上下関係に基づき、要素同士の近さ・遠さを見ることが可能

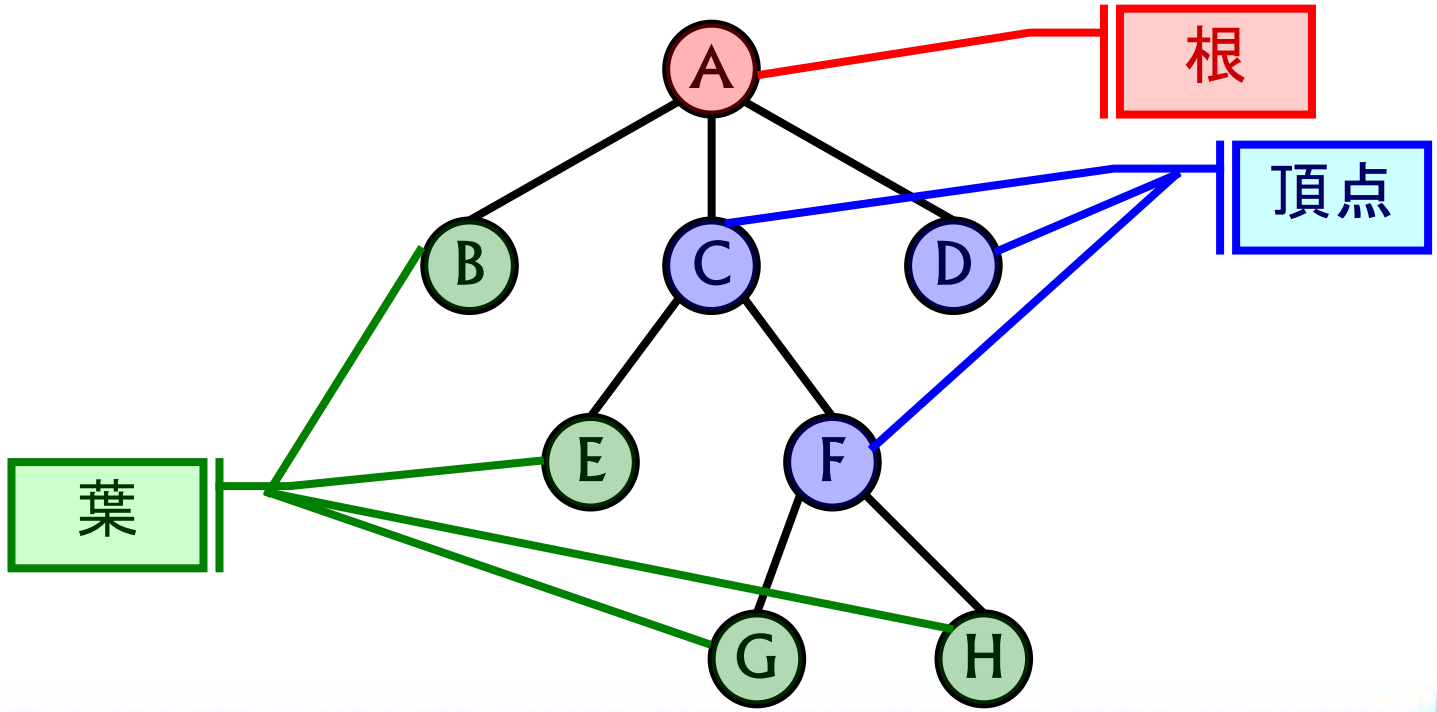
階層モデル[3](p. 147)

- 木構造
  - 1つの根元から枝分かれしていく構造(一種の階層モデル)
  - ただし、複数のノードから1つに向かって結ばれることはなし
    - 上のノードに直接つながる下のノードは0個以上(複数でもOK)
    - 下のノードと直接つながる上のノードは必ず1つ



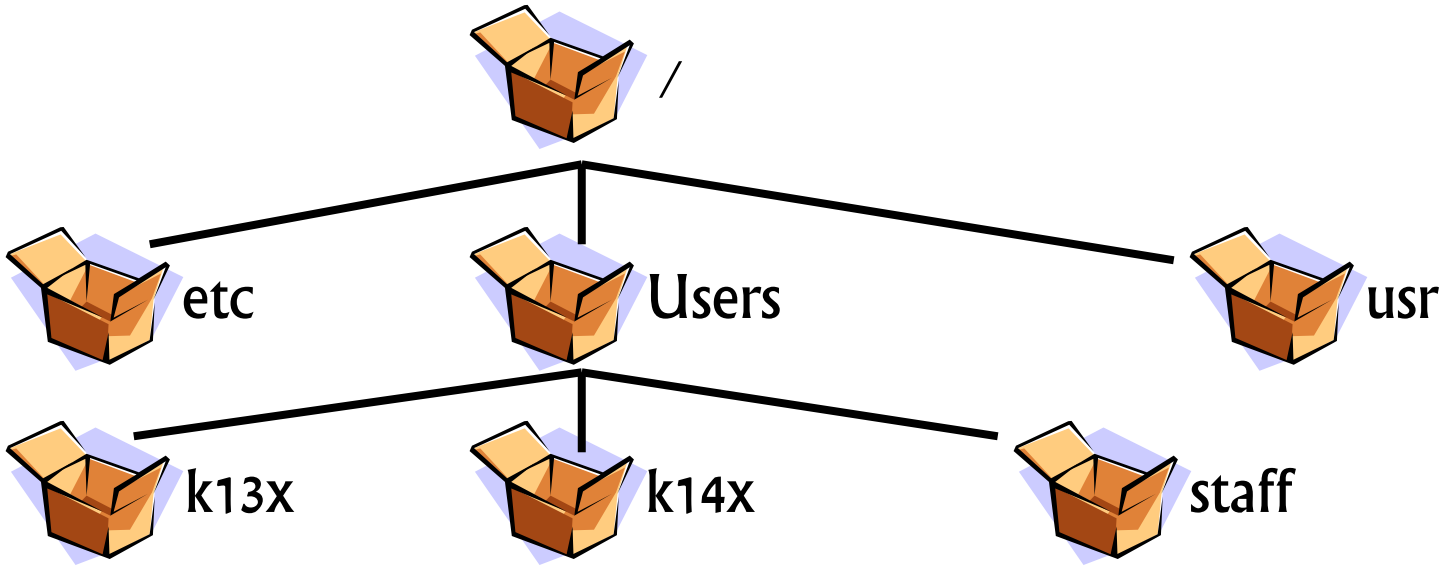
階層モデル[4](p. 147)

- 木構造
  - 1つ1つの項目: 頂点
  - 一番上の頂点: 根
  - これ以上枝分かれしていない頂点: 葉



階層モデル[4](p. 147)

- 例-ファイルシステム: 原則的に木構造

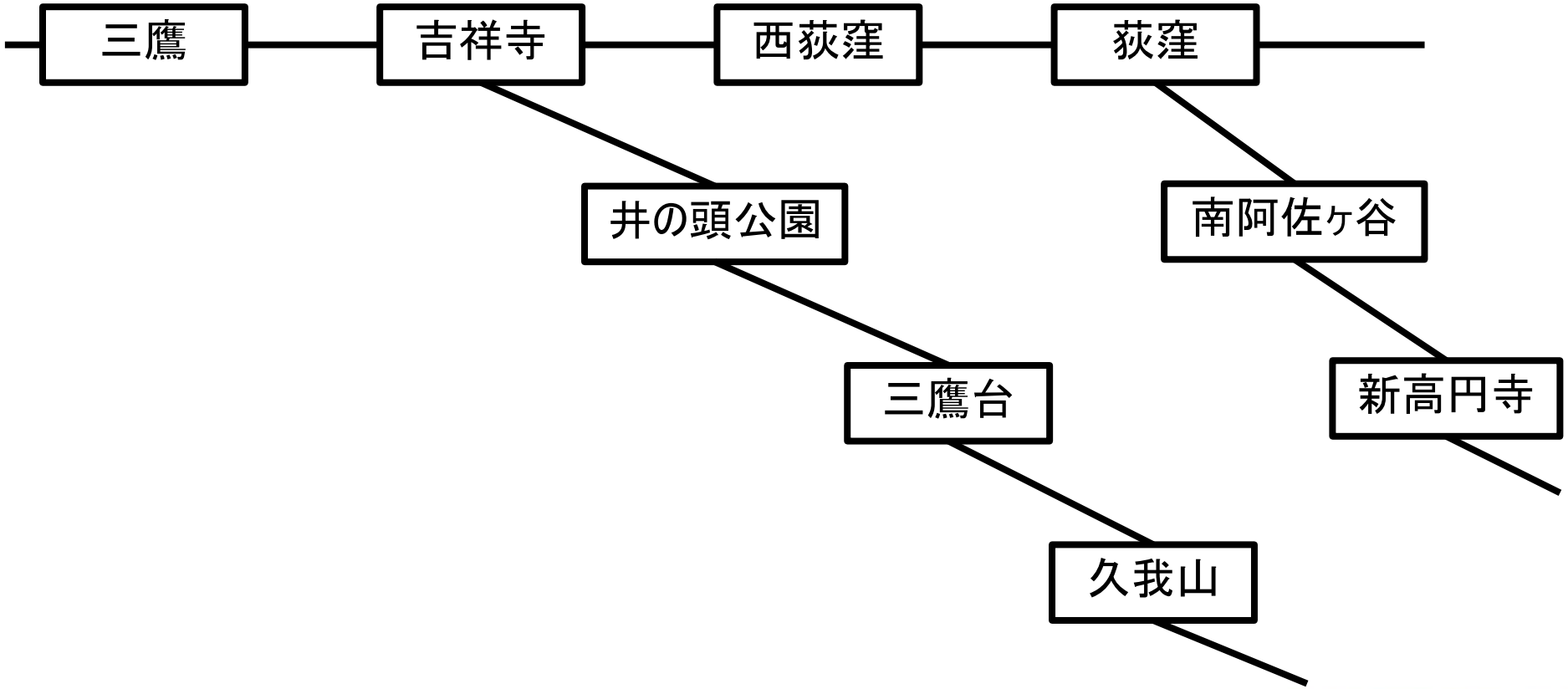


ネットワークモデル[1](p.149)

- 要素同士のつながり具合を表現したもの
  - 要素: ノード
  - ノードとノードを結ぶ線: エッジ
    - 矢印でないエッジ: 無向エッジ
    - 矢印のエッジ: 有向エッジ

ネットワークモデル[2](p. 149)

- 例: 路線図
  - 出発駅のノードからエッジをたどって目的駅までのルートを探索可能



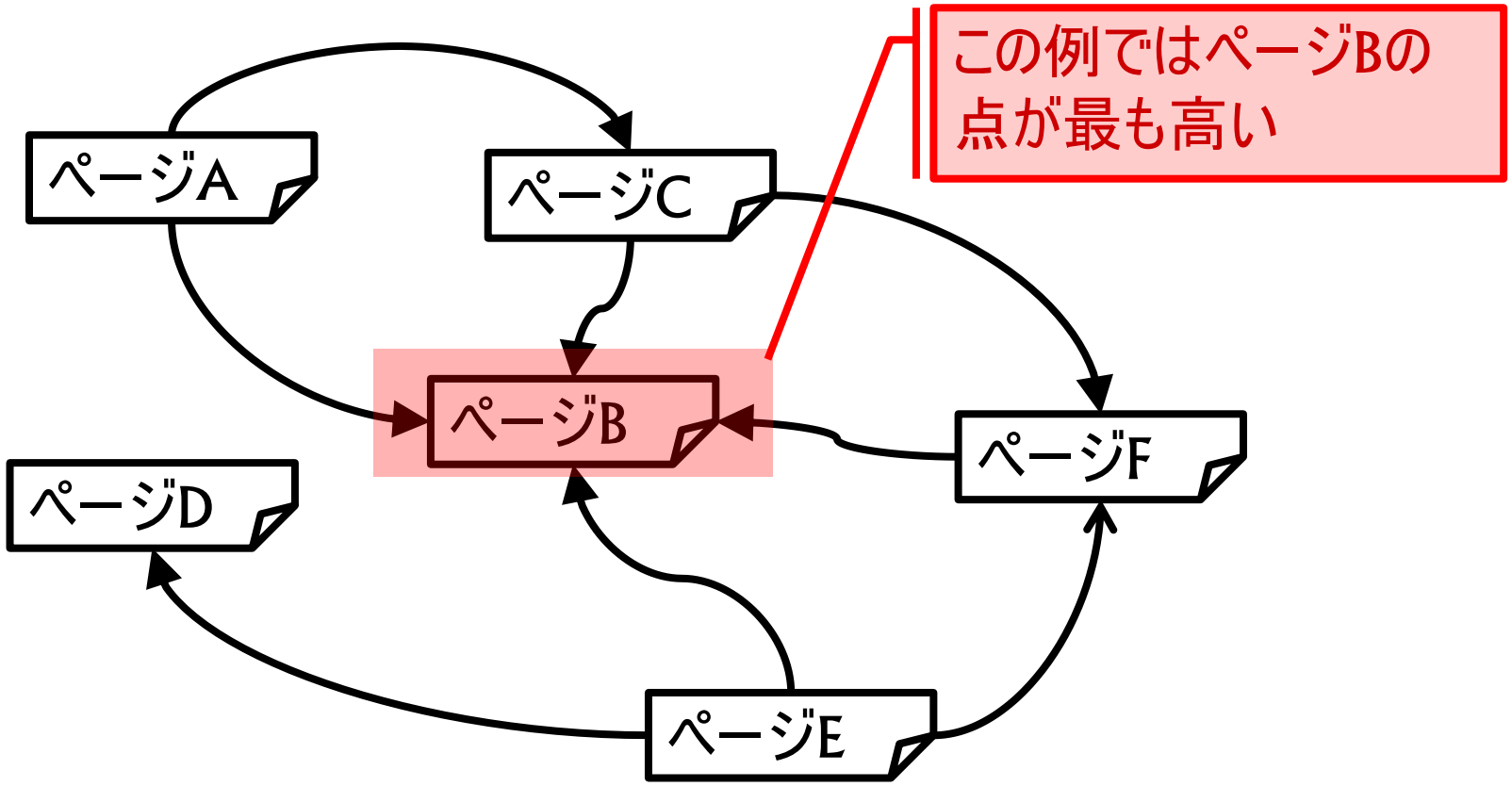
ネットワークモデル[3](p. 150)

- Webでのキーワード検索
  - クローリング: 世の中にどのようなWebページがあるかをあらかじめチェックすること
    - 検索のキーワードが入力されてから、目的のWebページを探すのでは多くの時間が必要
    - あらかじめWebページをチェックしてデータベース化しておくことで、検索の時間を削減
  - ランク付けの際にWebグラフを利用
    - ランク付け: 多くの検索結果に対して様々な観点で得点をつけ、得点を総合して、検索結果の表示の順序を決めること
    - Webグラフ: Webページをノード、リンクを有向エッジとしてWebページ同士のつながりを表したネットワークモデル

➡ 多くのWebページからリンクされているページは得点が高くなる

ネットワークモデル[4](p. 150)

⌘Webグラフ



※リンクだけでなく、キーワードがどの部分にあるか(タイトルや見出し、本文など)など、様々な観点をもとに
得点が計算され、それを総合して結果の表示順序が決定される。

関係モデル[1](p. 153)

- 表形式でデータを表したモデル
  - 関係: relation, リレーション

- 例: 住所録
  - 行(マスの横の並び)で1人分のデータ(レコード)を表現
  - 列(マスの縦の並び)でデータの項目(属性)を表現

氏名	住所	電話番号
東京子	東京都杉並区善福寺	03-1234-5678
善福寺花子	東京都武蔵野市吉祥寺	0422-98-5432

関係モデル[2](p. 153)

- コンピュータで扱う関係は様々な操作によって変化
  - 住所録の連絡先の追加・削除・変更, etc.
- 関係に対する操作
  - 和
  - 差
  - 射影
  - 選択
  - 直積

関係演算
(データベースでよく使われる)

関係に対する処理は、5つの関係演算を
組み合わせることで行われる

関係演算[和](p. 154)

- 複数のレコードをまとめる

東京子	東京都杉並区善福寺	03-1234-5678
善福寺花子	東京都武蔵野市吉祥寺	0422-98-5432

東京子	東京都杉並区善福寺	03-1234-5678
吉祥寺太郎	東京都中野区中野	03-9876-5432

東京子	東京都杉並区善福寺	03-1234-5678
善福寺花子	東京都武蔵野市吉祥寺	0422-98-5432
吉祥寺太郎	東京都中野区中野	03-9876-5432

関係演算[差](p. 154)

- 関係モデルからレコードを除く

東京子	東京都杉並区善福寺	03-1234-5678
善福寺花子	東京都武蔵野市吉祥寺	0422-98-5432

Ex.「東京子」のレコードを除く

善福寺花子	東京都武蔵野市吉祥寺	0422-98-5432

関係演算[射影](p. 154)

⌘指定された属性だけを抜き出す

東京子	東京都杉並区善福寺	03-1234-5678
善福寺花子	東京都武蔵野市吉祥寺	0422-98-5432

Ex.「名前」と「住所」の属性だけを抜き出す

東京子	東京都杉並区善福寺
善福寺花子	東京都武蔵野市吉祥寺

関係演算[選択](p. 154)

⌘指定された条件に合うレコードだけを集める

東京子	東京都杉並区善福寺	03-1234-5678
善福寺花子	東京都武蔵野市吉祥寺	0422-98-5432

「住所」の属性が「東京都武蔵野市吉祥寺」の選択

善福寺花子	東京都武蔵野市吉祥寺	0422-98-5432

関係演算[直積](p. 154)

- ある関係のレコードと別の関係のレコードをつないだレコードを全て集める

東京子	東京都杉並区善福寺	東京都杉並区善福寺	03-1234-5678
善福寺花子	東京都杉並区善福寺	東京都杉並区善福寺	03-9876-5432

東京子	東京都杉並区善福寺	03-1234-5678
東京子	東京都杉並区善福寺	03-9876-5432
善福寺花子	東京都杉並区善福寺	03-1234-5678
善福寺花子	東京都杉並区善福寺	03-9876-5432

※この場合、2つの関係がどちらも「住所」を持っているので、
「住所」の属性が一致するレコードを組み合わせる

Question!