アルゴリズムとデータ構造　第三回レポート課題

2020/06/28

US162039：梶田悠

2020

目次

[課題内容 1](#_Toc44594380)

[処理手順 2](#_Toc44594381)

[クイックソート 3](#_Toc44594382)

[ハッシュ構造 4](#_Toc44594383)

[ソースコード 4](#_Toc44594384)

[実行結果 4](#_Toc44594385)

[考察 4](#_Toc44594386)

# 課題内容

Wikipediaのデータを分析し、カテゴリに属するページタイトルを得るためのプログラムを作成する。 Wikipediaの各ページには，複数のカテゴリが付与してある。例えば、「file長良川」のページは「愛知県の河川」、「木曽川水系」、「名水百選」等といったカテゴリが付与されている。Wikipediaのページタイトル（例：「長良川」）と、そのページに付与されたカテゴリを記録したファイル（data.txt）と、全てのカテゴリ一覧記録したファイル（category.list）からは、ページに付与されているカテゴリは分かるが、あるカテゴリに属しているページが分からない。そこで，データを分析し、カテゴリに属するページタイトルを表示できるプログラムを作成した（sample.cpp）。このようなデータを作成しておくことで、カテゴリに属するページを高速に検索することが可能となる。 しかし、sample.cpp は非常に動作が遅く、処理が完了するまで約１時間程度の時間がかかる。そこで、データ構造にハッシュを採用することで処理を高速化させ、短時間（全て表示が終わるまで約１０秒）で処理が終了できるようにプログラムを改良せよ。

# 処理手順

## ハッシュの構築

コードにおいてハッシュから構築されたデータはhashManager変数である。addData関数で、引数でカテゴリとタイトルを渡すことでデータの追加をすることができる。

data.txtファイルからのデータを走査し、ページタイトルに対応する各カテゴリのペアを引数にaddData関数を実行してデータ構造に追加していく。addData関数内での処理は以下のようになっている。

1. カテゴリ名からハッシュ値を計算
2. ハッシュ値をインデックスにtable配列の要素を取得する（要素は線形リストになっている）
3. リストを走査し対象のカテゴリのノードを得る
4. ノードのメンバのタイトルが連続した文字列に引数で渡されたタイトルを結合する（この時同時にページ数をカウントするpageNum変数をインクリメントする）

## タイトルの検索

hashManagerはgetTitles関数でカテゴリを渡すと、対応するタイトルが連続した文字列を取得できる。category.listファイルの内容から全カテゴリを走査し、各カテゴリを引数にgetTitles関数を実行することで全カテゴリに対応するタイトル列を取得する。

getTitles関数内の処理は以下のようになっている。

1. カテゴリ名からハッシュ値を計算
2. ハッシュ値をインデックスにtable配列の要素を取得する（要素は線形リストになっている）
3. リストを走査し対象のカテゴリのノードを得る
4. ノードのメンバのタイトルが連続した文字列を返す

## 頻度順にソート

頻度順にソートするためにまずhashManagerにあるデータを一つのvectorに要素として集める。hashManagerにはtable配列の要素数を返すgetTableSize関数と、引数で渡した整数をインデックスとするtable配列の要素（先頭のノード）を返すgetListAt関数が実装されている。この二つを使ってhashManagerの全要素を走査し、各要素ごとにカテゴリとpageNumをメンバに持つCategoryクラスの変数をつくりvectorに追加していく。このカテゴリのデータが追加されているvectorはコードにおいてはcategories変数である。categoriesをsortCategories関数でソートする。sortCategories関数は前回のレポートのクイックソートをほとんどそのまま使用している。Categoryクラスで比較演算子のオペレータを定義していてpageNumによって比較されるようにしている為にクイックソート自体には特に今回の課題に依存した修正はしていない。

# ソースコード

作成したソースコードは[github](https://github.com/HarukaKajita/DataStructure_report03)にアップロードしています。見づらい場合や情報に不足がある場合は適宜参照して下さい。

|  |
| --- |
|  |

# 実行結果

実行結果 　report3.cppの実行結果の抜粋

|  |
| --- |
|  |

# 考察

　あ