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](#_Toc15502695)

[処理手順 2](#_Toc15502696)

[ソースコード 2](#_Toc15502697)

[実行結果 2](#_Toc15502698)

[考察 3](#_Toc15502699)

# 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

data.txt内の名前データのみを格納した配列と、名前と番号をデータとして持つノードから構成されるハッシュ構造を構築し、配列をクイックソートでソートししたのちに、配列内の名前を先頭から順に参照しハッシュ構造から同じ名前のノードを検索し番号を出力する事で課題を解決した。

課題の重要な部分であるクイックソートと、ハッシュ構造に関する部分についてのみ処理手順を解説する。

## クイックソート

　quick\_sort()関数は引数は、string配列の先頭アドレスとソート区間の両端のインデックスの3つである。配列の先頭要素の値を基準に処理を進める。先頭要素から末尾に向かって基準より大きい要素を探し、インデックス(=index1)を記録する。同様に末尾から、先頭に向かって基準より小さい要素を探しインデックス(=index2)を記録する。Index1<=index2であれば二つの要素を入れ替える。index1>index2になるまでこの処理を繰り返すと、配列が基準より小さい値が先頭側、大きい値が末尾側に集まる。この時、基準として使用した値は先頭要素の値なのでこの要素の位置を適切に調整する必要がある。先頭要素を基準より小さい値を持つ要素群の最後の要素と交換する事で、配列の大小関係の分割を完了する。基準の要素の前後の要素群をそれぞれ同じアルゴリズムで処理する事で書く要素が前後の要素と昇順の関係を持つようになり配列全体のソートが完了する。

## ハッシュ構造

　前述したように名前から番号をしたい状況なので、名前の文字列からハッシュ値を生成しこれをインデックスにtable配列にノードを格納した。別ノードとハッシュ値が競合した場合はチェイン法によって線形リストで同じハッシュ値のノードを管理するようにデータ構造を構築した。名前から番号を検索するsearch()関数は引数で入力された名前からハッシュ値を生成し該当する線形リストを順に同じ名前を持つノードが見つかるまで探索し、該当要素があればそのノードの番号のデータを出力するように実装した。

# 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 1　report3.cppの実行結果の抜粋

|  |
| --- |
|  |

# 考察

　あ