

進捗報告11/12

1. 現在取り組んでいること
文字列探索についての調査,可逆化?
2. 進捗状況
既存の有名な文字列探索についてまとめつつ, Cとjanusでの実装
3. 前回からの進捗
※プログラムでの進捗はgithub上に上げた
→<https://github.com/yokoyama-lab/reversible-algorithms>
力任せ法の2-aのプログラムが未だに未解決(読み込み続けるような状態, 無限ループに入っている可能性?)
 - ・前回の疑問点に関しては解決
→KMP法について理解した(文書等には書いていない)
→論文は読むべきか迷っている(Efficient string matching: an aid to bibliographic search)
 - ・Janusプログラムを記述してみた(kmp.j)
→compfのif文の `f[j] += i` 部分でエラーが出た

疑問点, 考察等

- ・教科書に載っていたプログラムで気になったこと
 - 各配列で一つ余分にスペースを確保しなければならない
 - テーブルの作成はkmp関数の中でする必要がある?
- ・文字列とパターンに使われている文字の種類が完全に一致するなら元の文字列に動かす距離を書き込めば可逆性を保持できるのではないか?

4. 今後の課題

- ・理解した部分を文字に起こす
- ・KMP法とBM法の可逆化
 - ・既存の有名なものを詳細にまとめ上げる
 - ・他にも多くのアルゴリズムも軽くでいいから列挙する

進捗報告11/3

1. 現在取り組んでいること
文字列探索についての調査,可逆化?
2. 進捗状況

既存の有名な文字列探索についてまとめつつ、Cとjanusでの実装

3. 前回からの進捗

※今回の進捗はすべてgithub上に上げた

→<https://github.com/yokoyama-lab/reversible-algorithms>

- ・ 前回の力任せ法のプログラムを試して実際に動くことを確認した
 - 前回のままではアサーションが間違っていて動かなかったので修正した
 - **2-aのプログラムだけ上手くいかなかった**（読み込み続けてるような状態、無限ループに入っている可能性？）
- ・ KMP法とBM法を教科書から引用してコミットした
- ・ 疑似コードの作成

疑問点等

- ・ kmp法の f の配列サイズの[MAXPAT]が何を表しているのか（max pattern?）
 - Efficient string matching: an aid to bibliographic search という論文に詳細が乗っているかもしれない
- ・ 動き自体は理解したが、テーブルの作成部分が理解しきれていない

4. 今後の課題

- ・ KMP法とBM法の可逆化
 - ・ 既存の有名なものを詳細にまとめ上げる
 - ・ 他にも多くのアルゴリズムも軽くでいいから列挙する