

進捗管理・報告(2020/5/8)

1. 現在取り組んでいること

- ・ 流れの可視化（卒業論文の続き）研究
- ・ 可逆コンピューティング研究

2. 進捗状況

流れの可視化

- ・ 研究内容をまとめたHTMLの作成
(<http://tetsuo.jp/lab/seminar/2018/tfda.html>)
- ・ 卒業論文の修正

可逆コンピューティング

- ・ 可逆コンピューティングについてと、この分野で行われてきたことの調査
(<https://www.overleaf.com/project/5e731b5059008200012799bd>)

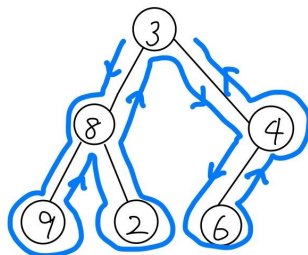
3. 前回からの進捗

可逆コンピューティング

- ・ 増田先輩の実装した可逆深さ優先探索アルゴリズムの動き方について理解した

```
procedure st(int in[], int k, int f, int visit)
  if visit != -1 then
    local int t = visit
    call st(in,k,f,in[t][1])
    call st(in,k,f,in[t][2])
    if in[t][0] = k then
      f ^= 1
    fi in[t][0] = k
  delocal int t = visit
fi visit != -1
```

例：



青線：計算順序

1. 根から探索を開始
2. 左の子を探索
3. 子を持たない時は戻って右の子を探索

特徴：

- ・ 必ず全ての要素を探索する（根から再帰を始めるため）
- ・ 根から再帰を始めるのは、根で探索を終了する必要があるため
→探索が終了したときの位置を一意にするため(逆計算のために必要?)
- ・ 要素の回数再帰されるので、空間計算量は $O(n)$
- ・ ゴミ出力量は0

改善点：

- ・ 空間計算量 $O(1)$ のアルゴリズムの実装
 - 探索した方向を記憶し、直前の要素の場所を一意に定めることができれば可能
 - 先生の作成したアルゴリズム

4.今後の課題

可逆コンピューティング

- ・ 調査の続き
 - Reversibility for efficient computing という論文が、可逆コンピューティングについて詳しくまとめているので、引き続きこれを読む。
- ・ 先生が作成した空間計算量が $O(1)$ のアルゴリズムについて、実装方法を理解する