



アルゴリズムとデータ構造

Group Work

スケジュール概要

- **第1回(12/1)**

- 準備 & 開発内容の説明 & グループで作業
 - コースナビ「データ等」のdat.tar.gzをDL.
- 計画書を提出

- 第2回～3回(12/15, 12/22) : 特別講義
- 中間計測(プログラム提出 ✕ 切: 12/20)
- 第4回(2018/1/12): グループで作業
- 最終成果物(プログラム)提出 ✕ 切(1/12)
- 発表資料提出 ✕ 切(2018/1/16)
- 第5回(2018/1/19): 成果発表会
- 報告書提出 ✕ 切(2018/1/26)

評価方法

- グループ全体の評価

- 成果物の性能

- 成果発表の質

- 個別の評価

- グループワーク中の各人の貢献度

評価方法

- グループ全体の評価

- 成果物の性能

- 教員が計測 → 成果発表会の日に公開

- 成果発表の質

- 皆さんが互いに評価, 教員の評価

- 個別の評価

- グループワーク中の各人の貢献度

- 皆さんが互いに評価
 - 計画書, 報告書

評価方法

● グループ全体の評価

○ 成果物の性能

● 教員が計測 → 成果物の性能

○ 成果発表の質

● 皆さんが互いに評価, 教員の評価

評価のポイント:

- ・ 発表はわかりやすかったか？
- ・ 提案手法は妥当だったか？
- ・ 提案手法を正しく実装できていたか？
- ・ 結果の分析は妥当か？

- 各人は, コースナビの“発表評価”から**自分が所属していないグループの中で特に良かった発表に投票する**. (8人のグループと7人のグループで1票の重みに差がありますが, 評価の際には正規化します.) 自グループへの投票は無効.
- ・ また, なぜ投票したのかコメントも書く.
- ・ コメントは集計後, 皆さんにシェアします.

評価方法

● グループ全体の評価

○ 成果

● 教

○ 成果

● 皆

- 各人は、コースナビの“作業評価”からメンバーの貢献度を評価します。
- 貢献度が高いと感じた上位4人を順位をつけて選んでください。
- 自分を選んでもOKです。

● 個別の評価

○ グループワーク中の各人の貢献度

- 皆さんが互いに評価
- 計画書, 報告書



コースナビ利用の練習

- 発表評価(練習)
- 作業評価(練習)

計画書の提出(本日×切)

- 以下の内容を含む文書を作成して、グループの代表者がコースナビ「計画書提出」に提出。
 - グループ名
 - メンバー
 - 作業の目的
 - 作業計画
 - 例)スケジュール表, 作業項目の列挙など.
 - メンバーの役割分担
 - 例)〇〇実装, データ分析, 発表資料作成, 議論, 記録係, 等...
- 分量:A4サイズで1~2枚程度でOK(必要に応じて増量して構わない.)

成果物提出のルール

- 提出するプログラムはC言語で記述すること.
- 提出時に利用してよいコンパイルオプションは
-lm, -O1, -O2, -Wall (wojと同ルール)
- 使用メモリの上限は2Gbyte.
- 実行ファイル名は「grpwk」とすること.
- Makefileも作ること.
- 計測用のテンプレートを使用すること. (template.c)
 - main_prgは別ファイルに移動してもOK. template.c以外のファイルを用意してもOK.
- 提出:
 - コンパイルに必要なファイルを一つのフォルダにまとめ, tarでアーカイブしてコースナビの「成果物提出」にグループの代表者が提出.
 - フォルダ名はグループIDとする事.
 - 例: グループIDが0の場合は「0.tar」を提出する.

性能評価

- 実行速度 (CPU時間) と精度 (後述) を評価指標とする.
- 実行速度と精度の順位の合計 s に対して, $\text{abs}(s-32)$ を各グループの基本スコアとする.
- 実行速度で順位が1位となったグループのスコアには「8」、2位のグループには「3」を加算. 精度の順位が1位, 2位の場合も同様.
- ただし, 「精度」に関しては, 最下位から数えて3番目までのグループにそれぞれ -10, -7, -2 のペナルティが与えられる.
- 中間計測に参加するグループのスコアには「1」を加算.
(コンパイルの不具合などを確認するためにも, 参加をお勧めします.)
- 例えば, 実行速度で5位, 精度で1位, 中間計測に参加した場合のスコアは, $\text{abs}(5+1-32)+8+1=35$

中間計測

- 参加の是非は自由.
- 2017/12/20, 23:59までに途中結果を提出したグループに関しては、本番と同様の方法で計測を行って、結果を公表します.
 - グループIDを知られたくない場合は、コードネームを使用可能なので、提出時に要望を記して下さい.
- 提出方法：本番の時と同じフォーマットでコースナビの「中間計測用提出」からグループの代表者が提出.

成果発表

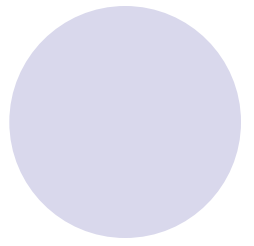
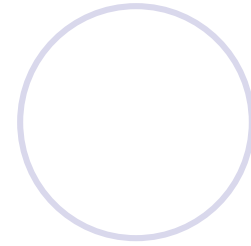
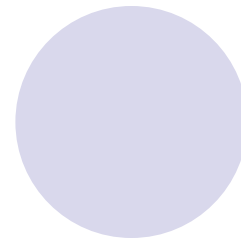
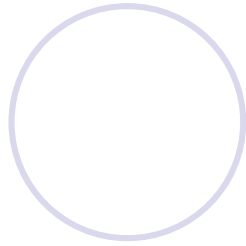
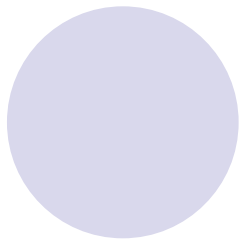
- 成果物に関する発表をする。
 - どのような方針で取り組んだのか？
 - 方針を実現するためにどのような方法論を用いたのか？
その方法論を用いた根拠は？
 - 実際にそれはうまくいったのか？
 - うまくいった(若しくはうまくいかなかった)要因の分析など
- グループの代表者(複数人でも可)が2分で発表
- グループの全員がポスター発表(60～90分を予定)
- 発表資料は1/16までにコースナビの「成果発表会用資料提出」にグループの代表者が提出。(通常のスライドで発表されることを想定していますが、動画などを利用するのもOKですので、その場合は事前にご相談ください。)

報告書の提出

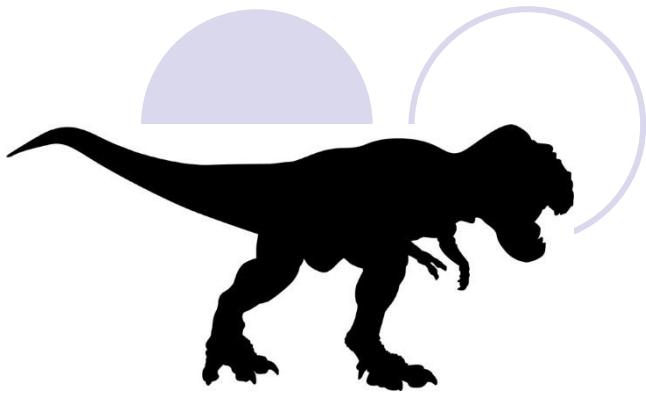
- 以下の内容を含む文書を作成し、2018/1/26までにグループの代表者がコースナビの「報告書提出」に提出。
 - グループ名
 - メンバー
 - 作業の目的と結果
 - 実際に行った作業（作業計画と対比させてもよい）
 - メンバーが実際に分担した役割
 - ディスカッションの議事録要約
 - 考察
 - 作業を進める上で難しかったこと、またそれをどうやって解決したか。
 - 今後同様の取り組みをする際には、どのような改善が望めるか。 など。
- 分量：A4サイズで1～2枚程度でOK（必要に応じて増量して構わない。）

提出〆切 & 作業スケジュール

- 2017/12/1, 23:59
 - 提出物: 計画書
 - 提出先: コースナビ「計画書提出」
- 2017/12/20, 23:59 (オプション)
 - 提出物: 中間計測用のプログラム群(tarファイル)
 - 提出先: コースナビ「中間計測用提出」
- 2017/1/12, 23:59
 - 提出物: 最終評価用のプログラム群(tarファイル)
 - 提出先: コースナビ「成果物提出」
- 2018/1/16, 23:59
 - 提出物: 発表資料
 - 提出先: コースナビ「成果発表会用資料提出」
- 2018/1/19, 23:59
 - 授業時間中の作業: 発表評価(コースナビの「発表評価」)
- 2018/1/26, 23:59
 - 提出物: 報告書
 - 提出先: コースナビ「報告書提出」
 - 作業: 作業評価(コースナビの「作業評価」)

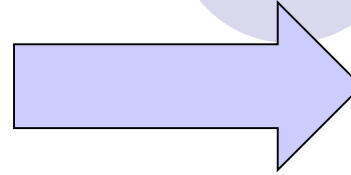


課題：古代生物の復元

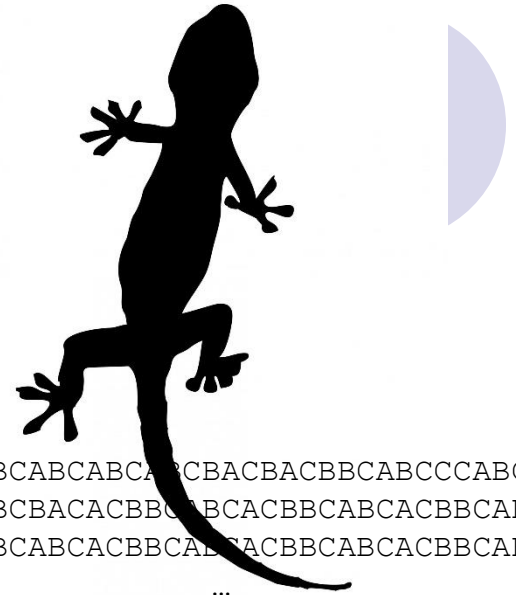


ABCABCABCABCABCACBACBACBBCABCCABC
ABCBBCBACACBBCABCACBBCABCACBBCAB
CACBBCABCACBBCABCACBBCABCACBBCAB

...

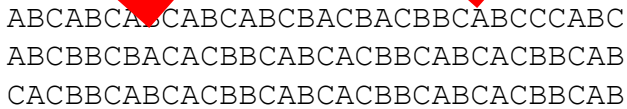


進化



ABCABCABCABCABCACBACBACBBCABCCABC
ABCBBCBACACBBCABCACBBCABCACBBCAB
CACBBCABCACBBCABCACBBCABCACBBCAB

...



A stylized illustration of a dinosaur skeleton, likely a T-Rex, shown in profile. The skeleton is brown and set against a white background. It features a large head with a prominent snout, sharp teeth, and three dark spots on the side of the head. The body is robust with a long, segmented tail and powerful legs.

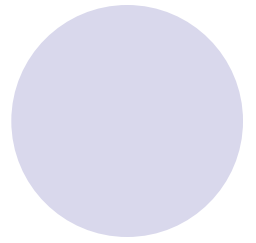
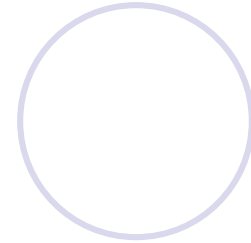
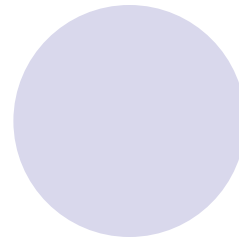
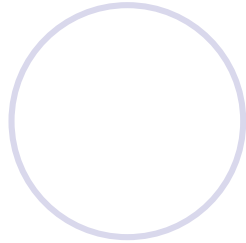
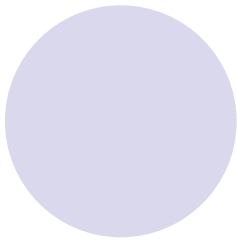
...

ABCABCABCABCABCACBACBACBACBCCABC
ABCBBACACBBACBACBACBACBACBACB
CACBBACACBACBACBACBACBACBACBACB

18

- 文字列 $T = t_1, t_2, \dots, t_N$, ($t_i \in \{a, b, c\}$) と, T に対して 文字の「挿入」, 「削除」, 「置換」の編集が加えられた T' が与えられたとする.
- ただし, T' における「挿入」, 「削除」の割合は 5%以内とする.
- 文字列 T のエラー入り部分文字列のセットを S とする. (本来の部分文字列に対して文字の「置換」が含まれている.)
- 目的:
 T' と S が与えられたとき, できるだけ速く, そして精度よく元のデータ(T)を復元する.

例



T = bacbaacababcbabacbcacabacaaabbbcbbaccbcbaccbbacbc

T' = bacbaacababcbabacbcacabbcaaabbbcbbacbacbcbaccabacbc

S = {
cabacbcbac
acababcbab
abacbcacab

...

}

性能評価

● 実行速度

○ 計測環境(予定)

- OS: Ubuntu 16.04LTS

- gcc: v5.4.0

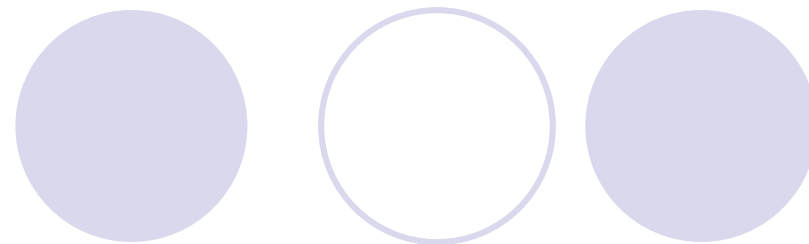
- CPU: Xeon E52643-v3 or Core i7-6700K

○ 10分で打ち切り

● 精度

○ T と提出物 X の編集距離

データに関して



● サンプルデータ

○ コースナビから入手可能

- datx_in: 入力データ ($x=0, \dots, 4$)

- datx_ref: 正解データ

○ 文字列の長さ: 約 10^5

○ 部分文字列: 長さ25 が20000本

● 本番用のデータ

- サンプルデータを生成したのと同じモデルを用いて作成. (つまり, サンプルデータと同質のデータを使って評価. 文字列長は同じ.)

入力データフォーマット

- 標準入力
- $T' \langle \text{改行} \rangle s_1 \langle \text{改行} \rangle s_2 \langle \text{改行} \rangle, \dots, s_k \langle \text{改行} \rangle$
- 例:

```
Bacbaacababcbabacbcacabbcaaabbcbcabacbcbaaccabacbc  
cabacbcbac  
acababcbab  
abacbcacab
```


出力データフォーマット

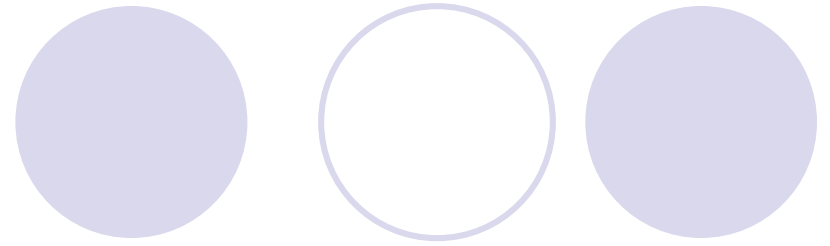
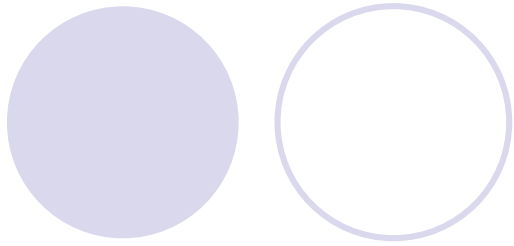
- 標準出力
- 復元した文字列＜改行＞

- 例：

`bacbaacababcbabacbcacabacaaabbcbcbbacbcbacccbcbacbc`

どんな方法で解くか？

- コピー？
 - 高速
 - 精度は悪い
- 全部編集距離を計算？
 - 精度はよい
 - 低速
- 高精度で高速な方法は？



- ぜひ活発な議論を

- ソースの共有

- github (<https://github.com/>)
- Dropbox
- Google Drive