生物情報ソフトウエア論Ｉ　（担当　森下）

試験日時　２０１５年６月４日　木曜日

１０：２５　～　１２：２５（２時間）

途中退室して構いません。

問題 1 (sorting)

```
void randomQuickSort1(int* target, int left, int right ) {
     if (left < right) {
          int i = partition(target, left, right);
               // i indicates the position of the pivot.
          randomQuickSort1(target, left, i - 1);
          randomQuickSort1(target, i + 1, right);
     }
}
```

(partition の実装は以下では変更しないため、プログラムを示していない)

プログラムを設計する際には、stack 領域を無駄に消費しないように、再帰的呼び出し (recursive call) を減らす工夫をするが、この観点からすると randomized quick sort は様々な問題をかかえている。

(1) この randomized quick sort の中には、再帰的呼び出しをしなくてよい部分がある。その部分を書き換えて tail recursion elimination を施したプログラムを示せ。

(2) partition による配列の分割がうまくゆかないと、stack を消費し尽してしまう現象 stack overflow が生じる。どのような分割が進行すると stack overflow が起こるか？これを避けるため、入力配列の長さが n の場合、プログラム実行中のいかなる時点でも、再帰的呼び出しが高々 $log_2 n$ 以下しか生じないようにプログラムを変形せよ。

問題２（k 番目に小さな要素を見つける問題）

各要素が異なる入力配列（長さ n）における k 番目に小さな要素を見つけるアルゴリズム SELECT の計算性能を考える。

SELECT

A) 入力配列を 5 つの要素からなるグループへ分割。残った要素がある場合、１つのグループにする。グループ数は $\lceil n/5 \rceil$ 個。ただし $\lceil x \rceil$ は実数 x 以上の最小の整数で、たとえば$\lceil 27/5 \rceil = 6$。

B) 各グループの中央値（median）を計算。

C) $\lceil n/5 \rceil$ 個の中央値全体の中央値 x （median of medians）を、SELECT を再帰的に実行して計算。ただし、$\lceil n/5 \rceil$ が奇数のとき中央値は１つに定まるが、偶数の場合は中央に２つの候補があるため、昇順で $\lceil n/5 \rceil/2$ 番目の値を中央値と定義する。奇数、偶数どちらの場合も、昇順で$\lceil \lceil n/5 \rceil/2 \rceil$番目の値が中央値となる。

D) x 未満と x 以上のブロックへ分割し、k 番目の要素があるブロックを決定。

E) k 番目の要素があるブロックを SELECT でさらに検索。

以下の設問に答えよ。

(1) x より大きい要素は $3n/10 - 6$ 個以上あることを証明せよ（同様に x より小さい要素は $3n/10 - 6$ 個以上ある）。よってステップ E で SELECT が検索するブロックの長さは高々 $7n/10 + 6$ である。

(2) 長さ n の入力配列を SELECT が処理する最悪時間計算を $T(n)$ と記述するとき、$T(n)$ が満たす条件式を示せ。ただしステップ A,B,D は線形時間で計算できると仮定してよい。

(3) n が十分に大きい時、$T(n)$ が $O(n)$ の関数であること、すなわち SELECT が線形時間アルゴリズムであることを示せ。

問題３　(suffix array, doubling)

記号列 S の i 番目の記号を $S[i]$ と表記する。$|S|$ は S の長さを示す。記号列および配列は 0-origin indexing で表現する。記号列 S は **$** で終わるものとし、**$** は他の位置には出現しないとする。記号列の辞書式順序を決める際には **$** はどの記号より小さいと約束する。次の各問に答えよ。

(1)　S = **CGGCGGCGGCT$**　に対する suffix array SA を示せ。

(2)　inverse suffix array ISA を示せ

(3)　SA を構築する Larsson-Sadakane の doubling algorithm とはなにか？　その動作を S = **CGGCGGCGGCT$**　を例として使って説明せよ。その際、$h$-group, h-rank, SA_h, approximate inverse suffix array ISA_h 等の記号を用いること。

(4)　$|S| = n$ のとき上の doubling algorithm の最悪計算量が $O(n \log n)$ であることを説明せよ。その際、問題２で述べた k 番目に小さな要素を見つけるアルゴリズムを用いて良い。

問題４　(suffix array, induced sorting)

induced sorting を利用して suffix array を線形時間で構築するアルゴリズムを設計する。次の各問に答えよ。

(1) $S[i]$ からはじまる suffix が、S[i+1]からはじまる suffix より辞書式順序に関して小さい（大きい）とき、記号 $S[i]$ と $S[i]$ からはじまる suffix を S-タイプ (L-タイプ) と呼ぶ。各 suffix が S もしくは L-タイプどちらであるかを $O(|S|)$ で判定するアルゴリズムを述べ、その動作を S = **CGGCGGCGGCT$** を例に説明せよ。

(2) $S[i]$ ($i > 0$) が S-タイプで、$S[i\text{-}1]$ が L-タイプの時、$S[i]$ を LMS (Left-Most S-type) 記号と呼ぶ。また LMS 記号からはじまる suffix を LMS suffix とよぶ。S の LMS suffix を示せ。すべての LMS suffix の辞書式順番が判明していると仮定して、suffix array を $O(|S|)$ で計算するアルゴリズムの動作を、S = **CGGCGGCGGCT$** を例に示せ。

(3) 長さ 2 以上の部分記号列 $S[i,k]$ において、$S[i]$ の後に最初に出現する LMS 記号が $S[k]$ となるとき、$S[i,k]$ を LMS prefix とよぶ。なお単一の LMS 記号も LMS prefix と呼ぶ。LMS prefix の辞書式順序を $O(|S|)$ で計算するアルゴリズムの動作を、S = **CGGCGGCGGCT$** を例に示せ。

(4) LMS prefix の辞書式順序から LMS suffix の辞書式順序を計算するアルゴリズムを簡潔に説明せよ。$|S| = n$ とするとき、上の induced sorting algorithm の計算量が $O(n)$ であることを説明せよ。

問題５　（Burrows-Wheeler Transform）

(1) 記号列 S の suffix array を SA、 Burrows-Wheeler Transform を BWT と表記する。S = **CGGCGGCGGCT$** の BWT を求めよ。

(2) BWT から元の記号列 S を再構築するために、

$$S[\,(|S|-2)-k\,] = BWT[\,T^k[0]\,] \quad (k = 0, \ldots, |S|-2)$$

を満たす写像 T を構築することを考える（$T^0[x]=x,\ T^{k+1}[x]=T[T^k[x]]$ ）。T を $O(n)$ 時間で計算するために、SA を利用する方法を説明せよ。SA を利用せずに T を $O(n)$ 時間で計算する方法を説明せよ。また S = **CGGCGGCGGCT$** のとき、対応する T を求めよ。

(3) 上記の方法を拡張し、問合せ記号列 W が出現する位置を $O(m)$ 時間で全て列挙するアルゴリズムの動作を、S = **CGGCGGCGGCT$** および W = **CGGC** を例にして示せ。また、そのアルゴリズムが使用する主記憶サイズを軽減する方法について述べよ。

問題 6（Seeded Alignment）

Seeded alignment とは異なる生物種の DNA 配列やアミノ酸配列の間で類似領域を探すために、しばしば使われる方法である。

(1) 長さ q の DNA 配列 Q と同じ長さの相同な領域 Q' をゲノム G 中に探索する問題を考える。Q と Q' の各塩基が一致する確率を M とする。Q と Q' が長さ $k\ (\leq q)$ の連続配列(seed) を共有する確率を sensitivity とよぶ。一方 Q と同じ長さのランダムな配列が長さ k の連続配列をたまたま共有する確率を偽陽性率と呼ぶ。sensitivity を上げ、偽陽性率を下げるにはどのような注意が必要か述べよ。

(2) Q と Q' の間に m 個のミスマッチがあり、完全マッチする最長の部分列の長さが k となる場合の数を $T[q][m][k]$ と表現する。Q と Q' が長さ k の連続配列を共有する sensitivity を、$T[q][m][k]$ と M を使って記述せよ。

(3) Q がゲノム G の部分配列である時、Q の mismatch tolerance とは何か述べよ。

(4) $q = 21$ のとき Q の mismatch tolerance が 3 であるか否かを調べたい。不完全マッチウインドウを利用して計算効率を上げるには、どのような不完全マッチウインドウ（長さとミスマッチ数）を設計すべきか、理由とともに述べよ。