平成25年度 東京大学大学院情報理工学系研究科 コンピュータ科学専攻 入学試験問題 専門科目 I

平成25年2月6日 10:00-12:30

注意事項

- (1) 試験開始の合図があるまで、この問題冊子を開けないこと.

 Do not open this problem booklet until the start of the examination is announced.
- (2) 3 題すべてに答えよ. 問題ごとに指定された解答用紙を使用すること.
 Answer the following 3 problems. Use the designated answer sheet for each problem.
- (3) 解答用紙および問題冊子は持ち帰らないこと.

 Do not take this problem booklet or any answer sheet out of the examination room.

下欄に受験番号を記入すること. Write your examinee's number in the box below.

受験番号	No.
------	-----

問題 1

実行列 A の非零要素(値が 0 でない要素)の数を nnz(A) で表す。また, $n\times n$ 実行列 A で,ある正整数 c があって,A のどの行,どの列についても,その行(列)に含まれる非零要素の数が c 以下であるような行列全体を Sp(n,c) と書くことにする。以下 C 言語では,実数を double で,整数を int で表現せよ。以下の問いに答えよ。

- (1) $n \times n$ の実行列ひとつを格納するための、メモリ使用量が O(n + nnz(A)) であるようなデータ構造をひとつ考え、C 言語で記述せよ。またその使い方を示すために、今答えたデータ構造で格納された行列 A と n 次ベクトル x に対し、y = Ax を効率的に計算するアルゴリズムを示せ。ただし x と y は長さ n の double の配列として格納されるものとする。
- (2) $A \in Sp(n,c)$, $B \in Sp(n,c)$ のとき, $A + B \in Sp(n,2c)$ であることを示せ.
- (3) $n \times n$ 行列 A, B は非零要素数が n^2 より十分小さいとして, C = A + B を効率的に計算する アルゴリズムを示せ. また, その計算量を答えよ. A, B は問い (1) で答えたデータ構造で格 納されていて, C も問い (1) で答えたデータ構造に格納するものとする. 必要であれば, 中間 的なデータ構造を使ってもよい.
- (4) $A \in Sp(n,c)$, $B \in Sp(n,c)$ のとき、 $AB \in Sp(n,c^2)$ であることを示せ.
- (5) $n \times n$ 行列 A, B は非零要素数が n^2 より十分小さいとして, D = AB を効率的に計算するアルゴリズムを示せ.また,その計算量を答えよ.A, B は問い (1) で答えたデータ構造で格納されていて,D も問い (1) で答えたデータ構造に格納するものとする.必要であれば,中間的なデータ構造を使ってもよい.

Problem 1

For a real matrix A, let nnz(A) be the number of the non-zero elements (elements whose values are not zeros). Let Sp(n,c), where c is a positive integer, be the set of $n \times n$ real matrices A that satisfy the following property: the number of non-zero elements in any row or any column of A is not larger than c. When you write your answers in C language, use double for real numbers and int for integers. Answer the following questions.

- (1) Define a data structure to store an $n \times n$ real matrix A, and describe it in C language. Here the memory consumption of the data structure must be O(n + nnz(A)). Also, to explain the usage of the data structure just answered, show an algorithm that computes y = Ax efficiently, for an $n \times n$ real matrix A and a real vector x of length n. Here A is given in the data structure you answered, and x and y are stored in arrays of double of length n.
- (2) Show that, if $A \in Sp(n,c)$ and $B \in Sp(n,c)$, then $A + B \in Sp(n,2c)$.
- (3) Show an algorithm that computes C = A + B efficiently, for A and B being $n \times n$ real matrices with the numbers of non-zero elements much less than n^2 . Answer the computational complexity of the algorithm. Here A and B are given in the data structure you answered in Question (1), and C must be stored in the data structure you answered in Question (1). You can use temporary data structures when necessary.
- (4) Show that, if $A \in Sp(n,c)$ and $B \in Sp(n,c)$, then $AB \in Sp(n,c^2)$.
- (5) Show an algorithm that computes D = AB efficiently, for A and B being $n \times n$ real matrices with the numbers of non-zero elements much less than n^2 . Answer the computational complexity of your algorithm. Here A and B are given in the data structure you answered in Question (1), and D must be stored in the data structure you answered in Question (1). You can use temporary data structures when necessary.

問題 2

アルファベット $\Sigma = \{a, b\}$ を考える.

- (1) 次の正則言語を受理する有限オートマトンを与えよ. (決定性でも, 非決定性でもよい)
 - (a) $\{x \in \Sigma^* \mid x \ c \ aa \ が部分語として現れる \}$
 - (b) $\{x \in \Sigma^* \mid x \text{ の長さは5以上で, 最後から数えて5番目の文字は } a \text{ である} \}$
- (2) Σ上の言語

$$L_1 = \{a^n b^n \mid n \text{ は 非負整数 }\}$$

が正則でないことを証明せよ.

(3) Σ上の言語

$$L_2 = \{a^n b^m \mid n, m \text{ id } n \neq m \text{ を満たす非負整数 } \}$$

が正則でないことを証明せよ.

Problem 2

Let $\Sigma = \{a,b\}$ be an alphabet.

- (1) For each of the following regular languages, give a finite automaton that accepts it. The automata can be deterministic or nondeterministic.
 - (a) $\{x \in \Sigma^* \mid x \text{ contains } aa \text{ as a subword } \}$
 - (b) $\{x \in \Sigma^* \mid \text{the length of } x \text{ is no less than 5, and its fifth last character is } a \}$
- (2) Prove that the following language L_1 over Σ is not regular.

$$L_1 = \{a^n b^n \mid n \text{ is a nonnegative integer}\}$$

(3) Prove that the following language L_2 over Σ is not regular.

$$L_2 = \{a^n b^m \mid n, \ m \text{ are nonnegative integers satisfying } n \neq m\}$$

問題3

同じ文字列が**2回**,**連続して**繰り返される文字列のことを**繰り返し文字列**と呼ぶものとする.たとえば "REPEAT" を 2 回繰り返した文字列 "REPEATREPEAT" は繰り返し文字列である.また,ある文字列 T に含まれる最長の繰り返し文字列である部分文字列のことを T の最長繰り返し文字列とよぶものとする.たとえば,"LETSREPEATREPEATIT" の最長繰り返し文字列は "REPEATREPEAT" である.ここで,T[i..j] は,文字列 T 中のi 番目から始まり,j 番目で終わる長さ j-i+1 の部分文字列を表すものとする.また,以下の問い中の平均計算量は,T が,あるアルファベット Σ ($|\Sigma| \geq 2$) 上のランダムな文字からなっているものとして解析せよ.以下の問いに答えよ.

- (1) "ABABCBCABCBCBCBC" の最長繰り返し文字列を求めよ.
- (2) 適当な整数 i,r に対し,T[i..i+2r-1] が繰り返し文字列であるかどうかを調べるアルゴリズムを述べ,その最悪計算量と平均計算量を示せ.
- (3) 適当な整数 m,r に対し、 $i \le m \le i + r 1$ を満たし、T[i..i + 2r 1] が繰り返し文字列となる i が存在するかどうかを調べるアルゴリズムを述べ、その平均計算量を示せ.
- (4) 適当な整数 m に対し, $i \le m \le i + r 1$ を満たし,T[i..i + 2r 1] が繰り返し文字列となるような i, r が存在するかどうか,もしあればそのうち最長のもの (すなわち r が最大のもの) をひとつ求めるアルゴリズムを述べ,その平均計算量を示せ.
- (5) 以下の疑似コードを用いて FindRepeat(T,1,n) を求めれば、T の最長繰り返し文字列を求めることができる。ここで n は T の長さとする。このアルゴリズムの平均計算量を示せ。

```
FindRepeat(T, i, j) {
1:
2:
        if (i > j) {
3:
           return nil;
4:
        }
        m \leftarrow \lceil (i+j)/2 \rceil;
5:
        P \leftarrow \text{FindRepeat}(T, i, m-1);
6:
7:
        Q \leftarrow \text{FindRepeat}(T, m+1, j);
8:
        R \leftarrow \{i \leq m \leq i+r-1 \text{ である繰り返し文字列 } T[i..i+2r-1] \text{ があれば}
                  そのうち最長のもの. なければ nil. };
        if (P=Q=R=nil) {
9:
           return nil;
10:
         } else {
11:
           return \{P, Q, R \text{ oper } nil \text{ でないもののうち, 最長のもの. } \};
12:
13:
         }
14:
      }
```

Problem 3

We call a string constructed by repeating the same string twice, without inserting any string between them, a tandem repeat. For example, the string "REPEATREPEAT" is a tandem repeat, as it is constructed by repeating "REPEAT" twice. A longest one among the substrings of T that form tandem repeats is said to be a longest tandem repeat of T. For example, "REPEATREPEAT" is the longest tandem repeat of "LETSREPEATREPEATIT".

Let T[i..j] denote T's substring of length j-i+1 that starts from the i-th position and ends at the j-th position. For the average time complexity analyses in the following questions, assume that T is constructed by a sequence of random characters from a given alphabet Σ ($|\Sigma| \ge 2$).

Answer the following questions.

- (1) Show a longest tandem repeat of "ABABCBCABCBCBCC".
- (2) Show an algorithm that checks whether T[i..i + 2r 1] is a tandem repeat or not for given i and r, and show both the worst time complexity and the average time complexity of the algorithm.
- (3) Show an algorithm that checks whether there exists i such that $i \leq m \leq i + r 1$ and T[i..i + 2r 1] is a tandem repeat for given m and r, and show the average time complexity of the algorithm.
- (4) Show an algorithm that, for given m, checks whether there exists i and r such that $i \leq m \leq i+r-1$ and T[i..i+2r-1] is a tandem repeat, and compute a longest one (i.e., one with the largest r) among them if such tandem repeats exist. Show the average time complexity of the algorithm.
- (5) We can obtain a longest tandem repeat of T by computing FindRepeat(T, 1, n), following the pseudo code below. Here n is the length of T. Show the average time complexity of the algorithm.

```
FindRepeat(T, i, j) {
1:
2:
         if (i > j) {
3:
            return nil;
4:
5:
         m \leftarrow \lceil (i+j)/2 \rceil;
         P \leftarrow \text{FindRepeat}(T, i, m-1);
6:
         Q \leftarrow \text{FindRepeat}(T, m+1, i);
7:
         R \leftarrow \{ \text{ If there exist tandem repeats } T[i..i+2r-1] \text{ such that } i \leq m \leq i+r-1, 
8:
                   then a longest one among such. nil if no such exists. };
         if (P=Q=R=nil) {
9:
             return nil;
10:
11:
          } else {
12:
             return {A longest non-nil string among P, Q and R. };
13:
          }
       }
14:
```