2019 年度 10 月期入学 / 2020 年度 4 月期入学
京都大学 大学院情報学研究科
修士課程 知能情報学専攻 入学者選抜試験問題
(情報学基礎)

2019 年 8 月 5 日 9:00～11:00

【注意】
1. 問題冊子はこの表紙を含めて 9 枚ある。
2. 試験開始の合図があるまで中を見てはいけない。
3. 試験開始後、枚数を確認し、落丁または印刷の不鮮明なものがあれば直ちに申し出ること。
4. 問題は日本語と英語の両方で出題されている。**すべて解答しなさい。**
   F-1 (F-1-1, F-1-2)　線形代数、微分積分……1-4 ページ
   F-2 (F-2-1, F-2-2)　アルゴリズムとデータ構造……5-8 ページ
5. 特に指定のない限り、日本語または英語で解答すること。
6. 解答用紙に記載されている注意事項についても留意すること。

The Japanese version of this document is the prevailing and authoritative version;
the English translation below is provided for reference only

October 2019 Admissions / April 2020 Admissions
Entrance Examination for Master's Program
Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto University
(Fundamentals of Informatics)

August 5, 2019
9:00 - 11:00

NOTES
1. This is the Question Booklet in 9 pages including this front cover.
2. Do not open the booklet until you are instructed to start.
3. After the examination has started, check the number of pages and notify proctors (professors) immediately if you find missing pages or unclear printings.
4. Questions are written in Japanese and English. **Answer all the questions.**
   F-1 (F-1-1, F-1-2)　Linear Algebra, Calculus……Pages 1 to 4
   F-2 (F-2-1, F-2-2)　Algorithms and Data Structures……Pages 5 to 8
5. Write your answer in Japanese or English, unless otherwise specified.
6. Read carefully the notes on the Answer Sheets as well.

修士課程　情報学基礎　【線形代数、微分積分】　問題番号 | F - 1

F-1-1 と **F-1-2** それぞれ別の解答用紙を用いて解答すること。

F-1-1

設問 1　以下の 2×2 正方行列 $\mathbf{A}$ について考える。

$$\mathbf{A} = \begin{pmatrix} 4 & -2 \\ -2 & 7 \end{pmatrix}$$

このとき、以下の問いに答えよ。
(1) $\mathbf{A}$ の行列式を求めよ。
(2) $\mathbf{A}$ の逆行列を求めよ。
(3) $\mathbf{A}$ の固有値と対応する固有ベクトルをすべて求めよ。
(4) $\mathbf{A}$ は正定値行列か判定せよ。理由も述べよ。
(5) $\mathbf{A}$ を対角化せよ。

設問 2　任意の実対称行列について、以下の問いに答えよ。
(1) すべての固有値が実数となることを証明せよ。
(2) 異なる固有値に対応する固有ベクトルは直交することを証明せよ。

（次のページに続く）

F-1-1 と **F-1-2** それぞれ別の解答用紙を用いて解答すること。

F-1-2

設問 下記の問いに答えよ。

(1) 体積が一定な直円錐の側面積が最小となるのは、その高さと底面の半径の比がいくらのときか。

(2) 下記の積分を求めよ。

(2.1) $\displaystyle\int_0^1 \log(1+\sqrt{x})dx$

(2.2) $\displaystyle\iint_D (x^2+y^2)^{-2}dxdy, \qquad D=\{(x,y) : x^2+y^2 \geq 1\}$

Question is translated in English in the section below; this translation is given for reference only.

Use one answer sheet for each of **F-1-1** and **F-1-2**.

F-1-1

Q.1 We have a 2×2 square matrix $\mathbf{A}$ given by

$$\mathbf{A} = \begin{pmatrix} 4 & -2 \\ -2 & 7 \end{pmatrix}.$$

Answer the following questions.

(1) Calculate the determinant of $\mathbf{A}$.

(2) Calculate the inverse matrix of $\mathbf{A}$.

(3) Calculate all the eigenvalue(s) and the corresponding eigenvector(s) of $\mathbf{A}$.

(4) Judge whether $\mathbf{A}$ is positive definite or not. Explain why.

(5) Diagonalize $\mathbf{A}$.

Q.2 Answer the following questions regarding an arbitrary real symmetric matrix.

(1) Prove that all the eigenvalues are real numbers.

(2) Prove that eigenvectors corresponding to distinct eigenvalues are orthogonal to each other.

(continued on the next page)

Use one answer sheet for each of **F-1-1** and **F-1-2**.

F-1-2

Q. Answer the following questions.

(1) Consider a straight cone with a constant volume. What is the ratio of the height to the base radius that minimizes the lateral surface area?

(2) Compute the following integrals.

(2.1) $\displaystyle\int_0^1 \log(1+\sqrt{x})dx$

(2.2) $\displaystyle\iint_D (x^2+y^2)^{-2}dxdy, \qquad D=\{(x,y) : x^2+y^2 \geq 1\}$

F-2-1 と **F-2-2** それぞれ別の解答用紙を用いて解答すること。

F-2-1

設問　以下の擬似コードで記述された関数 func(A) は数値配列 A を昇順にソートする。A,B,L,R は数値配列であり、配列のインデックスは 0 から始まる。A[x:y] は A[x] から A[y-1] までの部分配列を表し、関数 len(A) は配列 A の長さを返す。A.append(z) は配列 A の末尾に数値 z を追加し、print(A) は配列 A の内容を出力し、floor(z) は z 以下の最大の整数を返す。例えば A=[1,2,3] のとき len(A)=3、A[0:2]=[1,2]、A.append(4) により A=[1,2,3,4] となる。このとき、以下の問いに答えよ。

(1) [a], [b], [c], [d] にふさわしいコードを答えよ。

(2) func([2,9,5,3,7,0,1,4]) を実行したとき、出力を出力順に答えよ。

(3) このソートアルゴリズムの時間計算量を示し、その根拠を述べよ。

```
 1  func(A){
 2      n = len(A)
 3
 4      if(n == 1){
 5          return A
 6      }
 7
 8      m = floor(n / 2)
 9      L = func(A[0:m])
10      R = func(A[m:n])
11
12      B = []
13      i = j = 0
14
15      while( a ){
16          if(i == len(L) and j < len(R)){
17              B.append( b )
18              j = j + 1
19          } else if(j == len(R) and i < len(L)){
20              B.append( c )
21              i = i + 1
22          } else if( d ){
23              B.append(L[i])
24              i = i + 1
25          } else {
26              B.append(R[j])
27              j = j + 1
28          }
29      }
30      print(B)
31      return B
32  }
```

（次のページに続く）

F-2-1 と F-2-2 それぞれ別の解答用紙を用いて解答すること。

F-2-2

設問　n 文字のテキスト文字列 ($text$) の先頭から順に、m 文字のパターン文字列 ($pattern$) を探す問題を考える。
たとえば、図 (a) のように、$text$ の位置 i から始まる文字列と $pattern$ "ABCA" を比較し、3 文字目が不一致であったとする。この時、あらかじめ $pattern$ の性質を調べておけば、$pattern$ を 1 つ右にずらしても照合することはなく、$pattern$ を 2 つ右にずらして、$text$ の位置 $i+2$ から比較すればよいことがわかる。一方、図 (b) のように、4 文字目で不一致であった場合には、$pattern$ を 4 つ右にずらして、$text$ の位置 $i+4$ と $pattern$ の先頭の比較から再開すればよい。

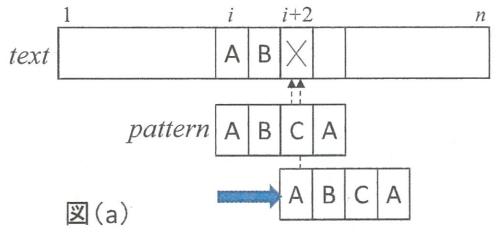


図 (a)

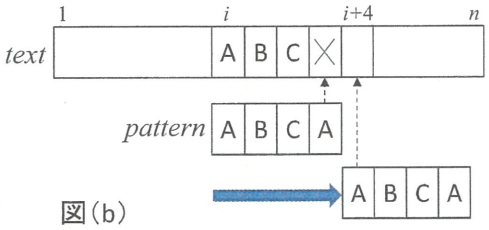


図 (b)

(1)　$pattern$ の位置 j で照合が失敗した時、$pattern$ を最大何文字まで右にずらせるかを $shift[j]$ で表すこととする。図 (a) では $shift[3] = 2$、図 (b) では $shift[4] = 4$ である。次の $pattern$ について、$shift[j]$ $(1 \leq j \leq 4)$ の値を求めよ。

(i) AAAB

(ii) ABAC

(2)　(1) の $pattern$ (ii) と $text$ "ABABBAABACA" との照合の過程を図示せよ。$pattern$ と $text$ のどの文字が比較されて行くかを明示すること。

(3)　このアルゴリズムの時間計算量を示せ。また、このアルゴリズムにおける比較の最大回数と、その具体例 ($pattern$ と $text$) を示せ。

Question is translated in English in the section below; this translation is given for reference only.

Use one answer sheet for each of **F-2-1** and **F-2-2**.

F-2-1

Q. The following pseudo code of a function `func(A)` performs sorting of a numerical array `A` in ascending order. In this code, `A`, `B`, `L` and `R` are numerical arrays whose indices start from 0. `A[x:y]` denotes `A`'s subarray going from `A[x]` to `A[y-1]` and `A.append(z)` appends a number `z` to the end of `A`. The function `len(A)` returns the length of `A`, the function `print(A)` outputs the content of the array `A`, and the function `floor(z)` returns the greatest integer less than or equal to `z`. For example, when `A=[1,2,3]`, `len(A)=3`, `A[0:2]=[1,2]` and `A.append(4)` turns `A` into `[1,2,3,4]`. Answer the following questions.

(1) Write appropriate code for [a], [b], [c] and [d].

(2) Write the outputs in the order of execution when `func([2,9,5,3,7,0,1,4])` is called.

(3) Describe the time complexity of the algorithm with the reason.

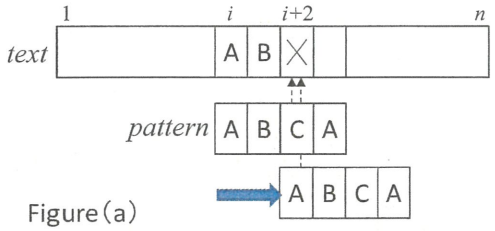
```
1  func(A){
2      n = len(A)
3
4      if(n == 1){
5          return A
6      }
7
8      m = floor(n / 2)
9      L = func(A[0:m])
10     R = func(A[m:n])
11
12     B = []
13     i = j = 0
14
15     while([ a ]){
16         if(i == len(L) and j < len(R)){
17             B.append([ b ])
18             j = j + 1
19         } else if(j == len(R) and i < len(L)){
20             B.append([ c ])
21             i = i + 1
22         } else if([ d ]){
23             B.append(L[i])
24             i = i + 1
25         } else {
26             B.append(R[j])
27             j = j + 1
28         }
29     }
30     print(B)
31     return B
32 }
```

(continued on the next page)

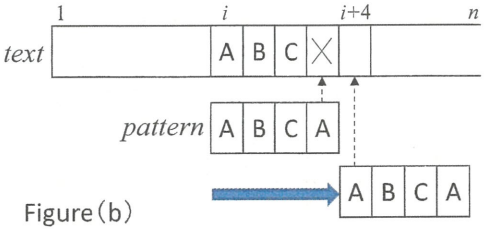
Use one answer sheet for each of F-2-1 and F-2-2.

F-2-2

Q. Consider a string matching problem of finding a *pattern* string with length m from the head of a *text* string with length n.
For example, in the case of Figure (a), we compare a string from the position i of *text* with *pattern* "ABCA", and find that the third character does not match. If the characteristic of *pattern* is examined beforehand, we notice that shifting *pattern* to the right by one character would be waste of time; it is more efficient to shift *pattern* by two characters and to restart the matching at the position $i + 2$ of *text*. On the other hand, in the case of Figure (b), when the fourth character does not match, we can shift *pattern* by four characters, and restart the matching between the position $i + 4$ of *text* and the first position of *pattern*.



Figure(a)



Figure(b)

(1) We use $shift[j]$ to denote the maximum number of characters by which *pattern* can be moved to the right when a mis-matching occurs at the position j of *pattern*. In the case of Figure (a), $shift[3] = 2$; in the case of Figure (b), $shift[4] = 4$. Calculate the values of $shift[j]$ $(1 \leq j \leq 4)$ for the following *patterns*:

(i) AAAB

(ii) ABAC

(2) Show the process of matching *pattern* (ii) of (1) with a *text* "ABABBAABACA". Clearly explain which characters in the *pattern* and the *text* are compared.

(3) Show the time complexity of the algorithm and the maximum number of character comparisons with a concrete example (*pattern* and *text*).