

筑波大学 情報学群 情報科学類・情報メディア創成学類

平成 28 年度 学群編入学試験

学力検査問題(専門科目)

[注意事項]

1. 試験開始の合図があるまで、問題の中を見てはいけません。
2. 解答用紙の定められた欄に、学群、学類(併願者は第一志望の学類)、氏名、受験番号を記入すること。
3. この問題冊子は全部で 11 ページ(表紙、白紙を除く)です。
4. 専門科目の選択について、
 - (ア) **情報科学類と情報メディア創成学類を併願する者**は、問題 1 から問題 6(数学、情報基礎、物理学)の計 6 問から 4 問を選択して答えなさい。ただし、情報メディア創成学類の合否判定においては、数学と情報基礎の解答のみを評価します。
 - (イ) **情報科学類を単願する者**は、問題 1 から問題 6(数学、情報基礎、物理学)の計 6 問から 4 問を選択して答えなさい。
 - (ウ) **情報メディア創成学類を単願する者**は、問題 1 から問題 4(数学、情報基礎)の計 4 問をすべて答えなさい。
5. 解答用紙は、専門科目で選択した 4 問に対して、各問 1 枚の合計 4 枚を用いること。
6. 解答用紙上部の 欄に解答する問題番号を記入すること。

問題 1 数学 (1)

(1) 関数 $f(x)$ が、 $x=a$ の近傍で C^2 級の関数であり、 $f''(a) \neq 0$ を満たすとき、以下の問いに答えなさい。

(1-1) $f(a+h)$ に、Taylor の定理を適用して展開しなさい。条件下において、できるだけ高い次数の項まで展開すること。また、剰余項の表記には θ_1 ($0 < \theta_1 < 1$) を用いること。

(1-2) $f(a+h) = f(a) + hf'(a+\theta h)$ ($0 < \theta < 1$) において、 $\lim_{h \rightarrow 0} \theta = \frac{1}{2}$ となることを示しなさい。

(2) 曲線 A: $y = x \cdot e^x$ について、以下の問いに答えなさい。

(2-1) 曲線 A、 x 軸 ($y=0$)、そして $x=a$ ($a < 0$) により囲まれる図形の面積 $S(a)$ を求めなさい。

(2-2) $\lim_{a \rightarrow -\infty} S(a)$ の値を求めなさい。

問題2 数学(2)

ベクトル a_1, a_2 は線形独立であるとする。このとき、以下の問いに答えなさい。

- (1) a_1, a_2 で張られる空間の直交基底 b_1, b_2 を求めなさい。
- (2) 次の連立一次方程式が解を持つための条件を示し、その条件を満すときの解 x を求めなさい。ここで、 b_1, b_2 は(1)で求めた直交基底であり、 b_1, b_2 および x は全て列ベクトルとする。

$$\begin{pmatrix} b_1 & b_2 \end{pmatrix} x = c$$

問題3 情報基礎 (1)

右の表のような a~z の 26 個の英小文字からなる文字列と正の整数の組を、効率よく格納・探索できるデータ構造とアルゴリズムを考える。ここでは、文字列をキー (key)、整数を値 (value) と呼ぶことにする。また、同じキーが複数格納されることはないものとする。

以下では、このキーと値の組を、配列を用いて格納・探索する方法と、木構造を用いて格納・探索する方法の二つについて、C 言語で書かれたプログラムによって示している。

以下の (1)~(5) に答えなさい。

キー	値
ant	120
any	260
anyway	110
apart	380
appear	290
apple	190
bad	400
bag	220
below	310
.	.
.	.

■ 配列を用いる方法

以下のような構造体 `struct kv` と大域変数の配列 `kv_array` を用いる。

```
#define N 10000

struct kv {
    char *key;
    int value;
};

struct kv kv_array[N];
```

今、この配列 `kv_array` に、10000 個のキーと値の組が、キーのアルファベット順に格納されているものとする。

以下の関数 `search_array` は、引数として与えられたキー `key` と組になっている値を、この配列を用いて求めるものである。なお、この関数は、与えられたキーと組になる値がない場合には -1 を返す。

この関数は、以下のような考え方に基づいたアルゴリズムを採用している。

- まず、配列 `kv_array` のちょうど真ん中の要素のキーと引数の `key` が一致するか調べる。一致すればその要素の値を返す。
- もし、配列 `kv_array` のちょうど真ん中の要素のキーよりも引数の `key` がアルファベット順で後ろにくるのであれば、配列 `kv_array` の前半分に `key` が格納されていることはあり得ないので、後半分だけを探索すればよい。

次ページに続く

- 逆に、配列 kv_array のちょうど真ん中の要素のキーよりも引数の key がアルファベット順で前にくるのであれば、配列 kv_array の後半分に key が格納されていることはあり得ないので、前半分だけを探索すればよい。
- このように、探す範囲を毎回半分に減らしていけば、効率良く探索することができる。

```
int search_array(char *key) {
    int start = 0;
    int end = N - 1;
    int middle;

    while (start <= end) {
        middle = (start + end) / 2;
        if (strcmp(kv_array[middle].key, key) == 0) {
            return kv_array[middle].value;
        } else if (strcmp(kv_array[middle].key, key) < 0) {
            start = middle + 1;
        } else {
            (a)
        }
    }

    return (b);
}
```

なお、関数 strcmp(s1, s2) は、二つの文字列 s1 と s2 をそれぞれの先頭の文字から 1 文字ずつ順に比較し、アルファベット順に並べたときに s1 が s2 よりも後ろにくるのであれば正の値を、s2 が s1 よりも後ろに来るのであれば負の値を返す。また両者が同じ文字列であれば 0 を返す。

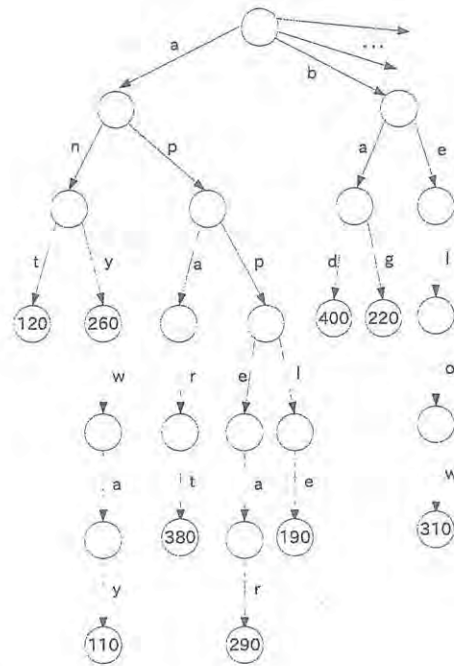
- (1) 上記の空欄 (a), (b) を埋めて関数 search_array を完成させなさい。
- (2) 配列 kv_array に格納されているキーと値の組の数を N 、引数として与えられるキー key の文字列長を k とするとき、関数 search_array の時間計算量を示し、その理由も述べなさい。

■ 木構造を用いる方法

右の図のような木構造でキーと値の組を表現する。各ノード間には、キーとなる文字列に含まれる文字を表す有向辺が付けられる。この図では、キーと組になる値は、それぞれのノードの中に示されている。

例えば、ルートノードから「a」「n」「t」という文字が付けられた三つの有向辺をたどると、120 という値が格納されているノードにたどり着く。これは、ant というキーと組になる値が120であることを表している。

同様に、「a」「n」「y」という文字が付けられた有向辺をたどることで、any というキーと組になる値が260 であることがわかる。



このような木構造を実現するため、以下の構造体を用いる。

```
struct node {
    int end_of_key;
    int value;
    struct node *next_char[26];
};
```

構造体 node は、図中の一つ一つのノードを表す。

end_of_key は、そのノードへの有向辺に付けられた文字がキーの文字列の最後の文字である場合には1が格納され、そうでない場合には0が格納される。例えば、上記の図の場合、120 や260 などの値が格納されているノードには end_of_key に1が格納され、そうでないノードには0が格納される。end_of_key に1が格納される場合には、そのキーと組になる(120 や260 などの)値が value に格納される。

配列 next_char は次のノードへのポインタが格納される。キーを構成する文字は a~z の26種類であるので、要素数を26としている。例えば、上記の図のルートノードから「a」「n」という二つの有向辺でたどりつくノードを n とすると、n の次の有向辺は「t」と「y」であり、これらはそれぞれ a~z の26種類の文字の中で20番目と25番目の文字なので、next_char[19] と next_char[24] に、次のノードへのポインタが格納される。また、次のノードはこの二つだけであるので、これ以外の next_char の要素には NULL が格納される。

以下の関数 search_tree は、ルートノードからこの木構造をたどって引数として与えられたキー key と組になっている値を求めるものである。なお、この関数は、

[次ページに続く](#)

与えられたキーと組になる値がない場合には-1を返す。

この関数の中で呼び出されている関数 numchar は、引数に与えられた文字が a~z の中で何番目の文字であるかの値を返す。例えば、numchar('a') は 1 を、numchar('b') は 2 を、numchar('z') は 26 を、それぞれ返す。

また、関数 strlen は引数に与えられた文字列の長さ (文字数) を返す。

```
int search_tree(struct node *root, char *key) {
    int i, len;
    len = strlen(key);
    struct node *n = root;

    for (i = 0; i < len; i++) {
        if (n->next_char[numchar(key[i]) - 1] == NULL) {
            return -1;
        } else {
            n = ;
        }
    }

    if (n->end_of_key == 1) {
        return ;
    } else {
        return -1;
    }
}
```

- (3) 上記の空欄 (c), (d) を埋めて関数 search_tree を完成させなさい。
- (4) 木構造に格納されているキーと値の組の数を N 、引数として与えられるキー key の文字列長を k とするとき、関数 search_tree の時間計算量を示し、その理由も述べなさい。
- (5) 本問ではキーに用いる文字は a~z の 26 種に限定していたが、例えば日本語のような文字種がとても多い場合には上記の struct node と同様のデータ構造では記憶効率が悪い。このことについて、
- その理由と、
 - 木構造を用いる方法の記憶効率を改善するにはどのようなデータ構造を用いるのが望ましいか、
 - その改善案の時間効率がどうなるか
- を、それぞれ簡潔に述べなさい。

問題4 情報基礎 (2)

図1(a)および図1(b)に示されるような有向グラフは、C言語の2次元配列を使って表せる。すなわち、グラフ G の頂点 P_i から P_j に向かう矢印が存在する時、2次元配列の $[i][j]$ 要素を1とし、それ以外の要素を0とする。このようにして作成した配列を行列とみなした場合、この行列をグラフ G の隣接行列と呼ぶ。例えば、頂点 P_0, P_1, \dots, P_4 と矢印からなるグラフ図1(a)を表す隣接行列は図2となる。

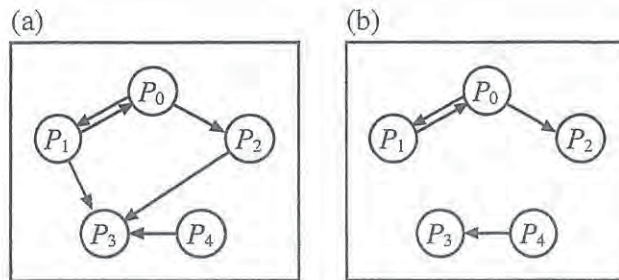


図1

	0	1	2	3	4
0	0	1	1	0	0
1	1	0	0	1	0
2	0	0	0	1	0
3	0	0	0	0	0
4	0	0	0	1	0

図2

N を頂点数、2次元配列 `int a[N][N]` を隣接行列とし、以下の問いに答えなさい。

- (1) 図1(b)を表す隣接行列を図2にならって書きなさい。
- (2) 配列 `a`が、図1(a)と図1(b)を表す隣接行列となっている場合のそれぞれについて、下に示されるC言語の関数 `maxranked()` の戻り値を答えなさい。

```
int maxranked() {
    int i, j, rank, max = 0, node = -1;
    for (i = 0; i < N; i++) {
        rank = 0;
        for (j = 0; j < N; j++) {
            if (a[j][i])
                rank++;
        }
        if (rank > max) {
```

[次ページに続く](#)


```

        max = rank;
        node = i;
    }
}
return node;
}

```

- (3) 上の関数 `maxranked()` の時間計算量を答えなさい。またその理由も答えなさい。なお、解答に際しては、グラフに存在する頂点の数を n 、矢印の数を m とし、これらを必要に応じて使いなさい。
- (4) 下に示される C 言語の関数 `is_reachable(int s, int g)` は、頂点 s から矢印をたどって頂点 g まで到達可能かどうかを判定し、到達可能な場合には 1 を、到達不可能な場合には 0 を返す関数である。空欄を埋め、この関数を完成させなさい。

```

int visited[N];
int is_reachable1(int s, int g) {
    int i;
    if (s == g)
        return 1;
    if (visited[s])
        return 0;
    visited[s] = 1;
    for (i = 0; i < N; i++) {
        
    }
    return 0;
}

```

```

int is_reachable(int s, int g) {
    int i;
    for (i = 0; i < N; i++)
        visited[i] = 0;
    return is_reachable1(s, g);
}

```

- (5) 下に示される C 言語の関数 `dijkstra(int s, int g)` は、頂点 s から矢印をたどって頂点 g まで到達可能な場合にはその 2 頂点間の最短距離を、到達不可能な場合には -1 を返す関数である。ただしここで有向グラフにおける 2 頂点 s および g 間の最短距離を、頂点 s から頂点 g へ到達するまでにたどる矢印の最小数とする。空欄を埋め、この関数を完成させなさい。

```

int dijkstra(int s, int g) {
    struct {

```

次ページに続く


```

    int found;
    int distance;
} node[N];

int i, j, min;
for (i = 0; i < N; i++) {
    node[i].found = 0;
    node[i].distance = N;
}

node[s].distance = 0;

for (j = 1; j < N; j++) {
    node[s].found = 1;
    if (s == g)
        return node[s].distance;

    for (i = 0; i < N; i++) {
        
    }

    min = N;
    for (i = 0; i < N; i++) {
        if (!node[i].found)
            if (node[i].distance < min) {
                min = node[i].distance;
                s = i;
            }
    }
}

return -1; /* unreachable */
}

```

問題5 物理学(1)

図1に示すように、ばね定数 k の N 個のばねと、質量 m の N 個の質点を交互に直列につなげ、天井から吊るした。ばねと質点は静止し、つり合っている。重力加速度を g とし、ばねは十分軽いものとする。以下の問いに答えなさい。

- (1) 図1のときのばねののびの総和を N, k, m, g を用いて表しなさい。
- (2) 図2に示すように、下端の質点に鉛直下向きの力を静かに加えていったところ、ばねののびの総和が x になり、つり合った。この時の力の大きさを N, k, m, g, x を用いて表しなさい。
- (3) 図1の状態から図2の状態まで、加えた力がした仕事 W を N, k, m, g, x を用いて表しなさい。
- (4) $N = 5, k = 98 \text{ N/m}, m = 100 \text{ g}, g = 9.8 \text{ m/s}^2, x = 25 \text{ cm}$ のとき、 W を求めなさい。ただし、単位は J とする。



図1

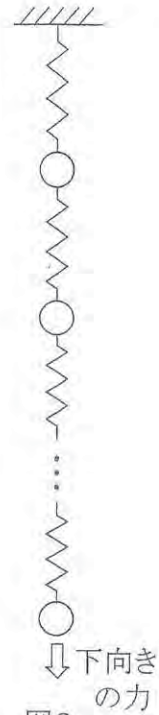


図2

問題 6 物理学 (2)

以下の問いに答えなさい。

- (1) 総電荷量が Q ($Q > 0$) で内部まで一様に帯電した半径 R の球がある (図 1 は、帯電した球の断面を表している)。球の中心からの距離を r として、以下の①と②の問いに答えなさい。なお、全ての領域で誘電率を ϵ_0 とする。

- ① 球の外部 ($r > R$) と内部 ($0 \leq r < R$) の電界の大きさ $|\vec{E}(r)|$ をそれぞれ求めなさい。
- ② 球の外部 ($r > R$) と内部 ($0 \leq r < R$) の無限遠点を基準とした電位 $V(r)$ をそれぞれ求めなさい。

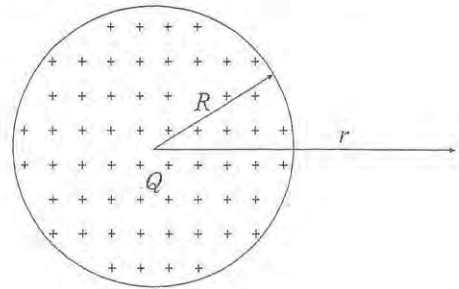


図 1

- (2) 抵抗網に関する以下の①と②の問いに答えなさい。

- ① 図 2 に示すように、起電力 E の電池に抵抗 $R_1 \sim R_4$ とスイッチ S を接続した。スイッチ S のオン/オフにかかわらず、同一の電流 I が流れるために抵抗 $R_1 \sim R_4$ が満たすべき条件を求めなさい。
- ② 図 3 に示すように、抵抗 R の導線を一方が $R-x$ 、もう一方が x になるように二つに分け、これら 2 つを並列接続した。そしてこの並列抵抗を、内部抵抗が r で起電力が E の電池に接続した。B を基準とした A-B 間の電位差 E_{AB} を求めなさい。さらに、 E_{AB} が最大になるときの x を求めなさい。

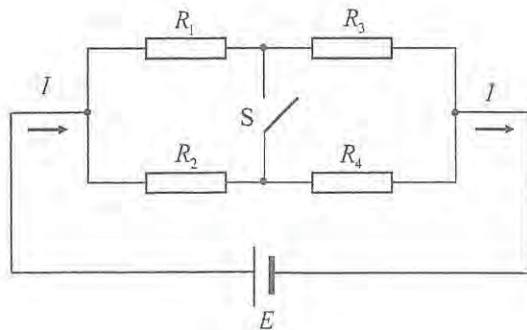


図 2

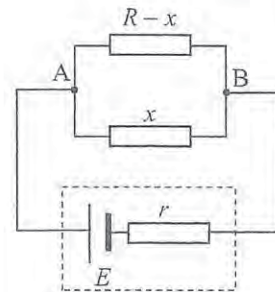


図 3