

## アルゴリズムとデータ構造 授業中練習問題14

次に示す「構造体に対するヒープソートの実現例」(教科書の List6-15 を構造体版に拡張)のある。このプログラムに関して、以下の問いに答えなさい。さらに、このプログラムを入力し、自分のパソコンでコンパイル、実行できることを確認してください。なお、プログラムの日本語部分は、英語、ローマ字に変更してかまいません。

```
/* 構造体に対するヒープソートの実現例 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define swap(type, x, y) do {type t; t = x; x = y; y = t;} while(0)

/*--- 会員データ ---*/
typedef struct {
    int no;          /* 番号 */
    char name[20];   /* 氏名 */
} Member;

/*--- 会員の番号の昇順比較関数 ---*/
int AscendingMemberNoCmp(const Member *x, const Member *y) {
    return x->no < y->no ? -1: x->no > y->no ? 1: 0;
}

/*--- 会員の番号の降順比較関数 ---*/
int DescendingMemberNoCmp(const Member *x, const Member *y) {
    return x->no < y->no ? 1: x->no > y->no ? -1: 0;
}

/*--- 会員の氏名の昇順比較関数 ---*/
int AscendingMemberNameCmp(const Member *x, const Member *y) {
    return strcmp(x->name, y->name);
}

/*--- 会員の氏名の降順比較関数 ---*/
int DescendingMemberNameCmp(const Member *x, const Member *y) {
    return strcmp(y->name, x->name);
}

/*--- 会員データ（番号と氏名）の表示（改行あり） ---*/
void PrintLnMember(const Member *x) {
    printf("%d %s\n", x->no, x->name);
}

/*--- 全データの表示 ---*/
void Print(const Member *data, int n) {
    int i;

    for(i=0; i < n; i++)
        PrintLnMember(data+i);
}

/*--- a[left]~a[right]をヒープ化 ---*/
static void updownheap(Member *a, int left, int right,
    int compare(const Member *y, const Member *z)) {

    Member temp = a[left]; /* 根 */
    int child;
    int parent;
```

```

for (parent = left; parent < (right + 1)/2; parent = child) {
    int cl = parent * 2 + 1; /* 左の子 */
    int cr = cl + 1;        /* 右の子 */
    child = (cr <= right && compare(a + cr, a + cl) > 0) ? cr : cl;
    /* 昇順なら大きい方, 降順なら小さい方 */
    if (compare (&temp, a + child) >= 0) ← ①
        break;
    a[parent] = a[child];
}
a[parent] = temp;
}
/*--- ヒープソート ---*/
void heapsort(Member *a, int n, int compare(const Member *y, const Member *z)){
    int i;

    for (i = (n - 1) / 2; i >= 0; i--)
        updownheap(a, i, n - 1, compare); ← ②

    for (i = n - 1; i > 0; i--) {
        swap(Member, a[0], a[i]);
        updownheap(a, 0, i - 1, compare);
    }
}
/*--- メニュー ---*/
typedef enum {
    TERMINATE, ASCEND_NO, ASCEND_NAME, DESCEND_NO, DESCEND_NAME, PRINT_ALL
} Menu;
/*--- メニュー選択 ---*/
Menu SelectMenu(void) {
    int i, ch;
    char *mstring[] = {
        "番号で昇順ソート", "名前で昇順ソート",
        "番号で降順ソート", "名前で降順ソート",
        "データを表示"
    };
};

do {
    for (i = TERMINATE; i < PRINT_ALL; i++) {
        printf("(%2d) %-22.22s ", i + 1, mstring[i]);
        if ((i % 3) == 2)
            putchar('\n');
    }
    printf("( 0) 終了 :");
    scanf("%d", &ch);
} while (ch < TERMINATE || ch > PRINT_ALL);

return (Menu)ch;
}
/*--- メイン ---*/
int main(void) {
    Menu menu;

```

```

Member data[] = {
    {1, "takahashi"}, {3, "konishi"}, {5, "ueda"}, {7, "sato"},
    {9, "niimi"}, {8, "okonishi"}, {2, "motoike"}, {4, "agemi"}
};
int ndata = sizeof(data)/sizeof(data[0]);

do {
    switch (menu = SelectMenu()) {
        case ASCEND_NO : /* 番号で昇順にソート */
            heapsort(data, ndata, AscendingMemberNoCmp);
            break;
        case ASCEND_NAME : /* 名前で昇順にソート */
            heapsort(data, ndata, AscendingMemberNameCmp);
            break;
        case DESCEND_NO : /* 番号で降順にソート */
            heapsort(data, ndata, DescendingMemberNoCmp);
            break;
        case DESCEND_NAME : /* 名前で降順にソート */
            heapsort(data, ndata, DescendingMemberNameCmp);
            break;
        case PRINT_ALL : /* 全データを表示 */
            Print(data, ndata);
            break;
    }
} while (menu != TERMINATE);
return 0;
}

```

- 1) このプログラムの動作直後に、数字の1を入力しました。このとき、次の問いに答えなさい。
  - (ア) この状態で、矢印②の for ループ中の関数 updownheap() が呼出された直後の構造体配列 a の値を、i の値毎に示しなさい。
  - (イ) この状態で、矢印②直前の for ループが繰り返されている間に、関数 updownheap() 中の矢印①の if 文が実行される回数は何回ですか。
- 2) このプログラムの動作直後に、数字の4を入力しました。このとき、次の問いに答えなさい。
  - (ア) この状態で、矢印②の for ループ中の関数 updownheap() が呼出された直後の構造体配列 a の値を、i の値毎に示しなさい。
  - (イ) この状態で、矢印②直前の for ループが繰り返されている間に、関数 updownheap() 中の矢印①の if 文が実行される回数は何回ですか。