

平成25年度 博士前期課程 入試問題
1 アルゴリズムとプログラミング 解答

平成25年6月28日

(1)

ア i--

イ MAX-i-1

(2)

(2-1)

(5, a) (2, d) (6, e) (4, h) (1, b) (8, f) (3, c) (7, g)

swap: 12

(1, b) (2, d) (3, c) (4, h) (5, a) (6, e) (7, g) (8, f)

(2-2) (a)

(1, a) (2, d) (4, e) (6, h) (8, b) (3, f) (5, c) (7, g)

swap: 6

(1, a) (2, d) (3, f) (4, e) (5, c) (6, h) (7, g) (8, b)

(2-2) (b)

(8, c) (2, f) (3, h) (4, g) (5, d) (6, e) (7, b) (1, c)

swap: 13

(1, c) (2, f) (3, h) (4, g) (5, d) (6, e) (7, b) (8, c)

(2-2) (c)

(8, c) (1, f) (2, h) (3, g) (4, d) (5, e) (6, b) (7, c)

swap: 7

(1, f) (2, h) (3, g) (4, d) (5, e) (6, b) (7, c) (8, c)

(2-2) (d)

(2, x) (4, y) (6, z) (8, w) (2, s) (4, t) (6, u) (8, v)

swap: 10

(2, s) (2, x) (4, t) (4, y) (6, u) (6, z) (8, v) (8, w)

(2-3)

エ aj <= ai

オ k

(3)

15 行目より、**key** が同値の場合でも入れ替え操作を行うため、
整列前の要素の並び順の前後関係は整列後に維持されない。

よって関数 **sort** が実現しているアルゴリズムは安定でない。

(上記 (2-2)(d) の結果より明らか)

リスト 1: q1.c

```
1 #include <stdio.h>
2
3 #define MAX 8
4
5 struct item_t {
6     int key; char val;
7 };
8
9 void sort(struct item_t data[])
10 {
11     int i, j, tmp_key;
12     char tmp_val;
13
14     for(i = 1; i < MAX; i++)
15         for(j = i; j > 0 && data[j-1].key >= data[j].key; j--){
16             tmp_key = data[j].key; tmp_val = data[j].val;
17             data[j].key = data[j-1].key;
18             data[j].val = data[j-1].val;
19             data[j-1].key = tmp_key; data[j-1].val = tmp_val;
20         }
21 }
22
23 void show(struct item_t data[])
24 {
25     int i = MAX;
26
27     while( i-- > 0)
28         printf("(%d,%c)", data[MAX-i-1].key, data[MAX-i-1].val);
29     printf("\n");
30 }
31
32 int main(void)
33 {
34     struct item_t data[MAX] = { {5,'a'}, {2,'d'}, {6,'e'}, {4,'h'}, {1,'b'}, {8,'f'}, {3,'c'}, {7,'g'} }; // 2-
35
36     sort(data);
37
38     show(data);
39
40     return 0;
41 }
```