

1 マージ・ソート

```
1  /*   マージソート (Merge Sort)   */
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  #define n 20
6  int b[n];
7
8  merge(int a[], int p, int q)
9  {
10     int i, k, l, r, m, left;
11     if (p < q){
12         m = (p+q)/2;
13         merge(a, p, m); /* 左半分のソート */
14         merge(a, m+1, q); /* 右半分のソート */
15
16         /*   併合 (merge) 開始   */
17         for (i = p; i <= q; i++) b[i] = a[i];
18
19         l=p; r=m+1;
20         for (k = p; k <= q; k++){
21             if ( l <= m && r <= q ) {
22                 if ( b[l] < b[r] ) left = 1;
23                 else left = 0;
24             }
25             else {
26                 if ( r > q ) left = 1; /* 右半分が空になったとき */
27                 else left = 0; /* 左半分が空になったとき */
28             }
29
30             if ( left == 1 )
31                 { a[k] = b[l]; l++; }
32             else
33                 { a[k] = b[r]; r++; }
34
35         }
36     }
37 }
38
39 main ()
40 {
41     int i,a[n];
42     srand(time(NULL));
43
44     for (i = 0; i<n; i++)
45         a[i]=rand();
46
47     for (i=0; i<n; i++) {
```

```

48     printf("%10d ",a[i]);
49     if ( i%5 ==4 ) printf("\n");
50 }
51
52 merge(a, 0, n-1);
53 printf("\n");
54 for (i=0; i<n; i++) {
55     printf("%10d ",a[i]);
56     if ( i%5 ==4 ) printf("\n");
57 }
58 }

```

実行例

```

1122894477 1658833269 1726213407 376530022 130828341
1928255780 536167434 2095300529 1131389460 733574219
1448833932 5326823 1516100676 216452583 1902806693
1598403710 2093372475 1783279246 414865579 1697490356

5326823 130828341 216452583 376530022 414865579
536167434 733574219 1122894477 1131389460 1448833932
1516100676 1598403710 1658833269 1697490356 1726213407
1783279246 1902806693 1928255780 2093372475 2095300529

```

2 2分探索

```

1 /*      Binary Search(1) */
2 #include <stdio.h>
3 #include <stdlib.h>
4 #define n 20
5 #include "selection_sort.c"
6
7 void B_search(int x[],int key)
8 {
9     int l=0,m,r=n-1;
10
11     while ( l<= r ) {
12         m=(l+r)/2;
13         if (key < x[m]) /* 左半分にある場合 */
14             r=m-1;
15         else if (key > x[m]) /* 右半分にある場合 */
16             l=m+1;
17         else { /* ちょうど真中にある場合 */
18             printf(" %d exists at %dth position.\n",key,m);
19             return;
20         }
21     }
22     printf(" %d does not exist \n",key);
23 }

```

```

24 main()
25 {
26     int x[n], key, i;
27     srand(time(NULL));
28
29     for (i=0; i<n; i++)
30         x[i]=rand() % 100;
31     selection_sort(x);
32
33     printf(" Input data \n");
34     for (i=0; i<n; i++) {
35         printf("%10d ",x[i]);
36         if ( i%5 ==4 ) printf("\n");
37     }
38
39     scanf("%d",&key);
40     B_search(x,key);
41 }

1  /*      Binary Search(2)
2          再帰呼び出し版
3  */
4  #include <stdio.h>
5  #include <stdlib.h>
6  #define n 20
7  #include "selection_sort.c"
8
9  void B_search(int x[],int key, int l, int r)
10 {
11     int m;
12     if ( l > r )
13         printf("%d does not exist.\n",key);
14     else
15         {
16             m = (l+r)/2;
17
18             if (key < x[m])      /* 左半分にある場合 */
19                 B_search(x, key, l, m-1);
20             if (x[m] < key)     /* 右半分にある場合 */
21                 B_search(x, key, m+1, r);
22             /* ちょうど真中にある場合 */
23             if (x[m] == key )
24                 printf(" %d exists at %dth position.\n",key,m);
25         }
26 }
27
28 main()
29 {
30     int x[n], key, i;

```

```

31     srand(time(NULL));
32
33     for (i=0; i<n; i++)
34         x[i]=rand() % 100;
35     selection_sort(x);
36
37     printf(" Input data \n");
38     for (i=0; i<n; i++) {
39         printf("%10d ",x[i]);
40         if ( i%5 ==4 ) printf("\n");
41     }
42
43     scanf("%d",&key);
44     B_search(x,key,0,n-1);
45 }

```

実行例

Input data

11	14	16	16	19
20	35	42	44	47
50	55	62	66	74
83	85	87	89	93

55

55 exists at 11th position.