クイックソートのプログラム

```
 1 #include <stdio.h>
 2 #include <stdlib.h>
 3 #define n 20
 4
 5 int x[n];
 6
 7 int partition(int a,int left,int right)
 8 {
 9   int pivot,i,j,tmp;
10
11 /*  x[right]; をキーにして左右に分割   */
12   pivot=x[right];
13   i=left-1;
14   j=right;
15
16   while(1){
17     do {
18       i=i+1;
19     } while(x[i]<pivot);
20
21     do {
22       j=j-1;
23     } while(x[j]>pivot);
24
25     if (i>=j){
26       break;
27     }
28     /*  x[i] と x[j] の交換  */
29     tmp=x[i];
30     x[i]=x[j];
31     x[j]=tmp;
32   }
33
34     /*  x[i] と x[right] の交換  */
35   tmp=x[i];
36   x[i]=x[right];
37   x[right]=tmp;
38
39   return (i);
40 }
41
42 void quick_sort(int a,int left, int right)
43 {
44   int i;
45   if (left<right){
46     i=partition(a,left,right);
47     quick_sort(a,left,i-1);
48     quick_sort(a,i+1,right);
```

```
    49   }
    50 }
    51
    52 void main()
    53 {
    54   int left,right,a,i,j;
    55   srand(time(NULL));
    56
    57   for (i=0; i<n; i++)
    58     x[i]=rand();
    59
    60   printf(" Input data: \n");
    61   for (i=0; i<n; i++) {
    62     printf("%10d ",x[i]);
    63     if ( i%5 == 4 ) printf("\n");
    64   }
    65
    66   a=x[n-1];
    67   left=0;right=n-1;
    68   quick_sort(a,left,right);
    69
    70   printf("\n");
    71   printf(" Sorted data: \n");
    72   for (i=0; i<n; i++) {
    73     printf("%10d ",x[i]);
    74     if ( i%5 == 4 ) printf("\n");
    75   }
    76 }
```

実行結果

```
 Input data
1318205241 1551501218 1696632230 1212552250 1104523632
1004608420  913769444 1978249790  595019623 1488637818
1110034874  775666943  847175045 1924699926  131765598
 501197845  467315184 1706839961 1536177720  774825336

 Sorted data
 131765598  467315184  501197845  595019623  774825336
 775666943  847175045  913769444 1004608420 1104523632
1110034874 1212552250 1318205241 1488637818 1536177720
1551501218 1696632230 1706839961 1924699926 1978249790
```

実行時間は，$n = 10,000,000$ に対してクイック・ソートは 4.74 秒，$n = 50,000$ に対して最小値選択法が 4.68 秒という結果が得られた。

問題 教科書にある 3 つのソーティングアルゴリズム (最小値選択法，バブルソート，挿入法) のプログラムを作り，計算時間を測定し，計算時間が実際にデータの個数 n の二乗に比例することを示せ。ここでデータの個数 n は $n = 2000, 4000, 6000, \cdots, 20000$ という順に 2,000 ずつ増やしていく。計算時間は time コマンドで測定する。

具体的には，プログラムをコンパイルした後

```
time a.out
```

と入力すると

```
12.780u 0.000s 0:12.79 99.9%     00k 0+0io 92pf+0w+
```

あるいは

```
real    0m12.790s
user    0m12.780s
sys     0m0.000s
```

と出力される。これは user のために使われた時間が 12.780 秒，システムのために使われた時間が 0.000 秒，合計 12.79 秒ということを表している。

以下にプログラムのヘッダ部分と乱数の発生部分だけを示しておく。

```
#include <stdio.h>
#include <stdlib.h>
#define n 10000

void main()
{
  int i,a[n];
  srand(time(NULL));     /* 乱数の初期化 */
  for (i=0; i<n; i++)    /* 整数値乱数を n 個発生させる */
    a[i]=rand();
     <プログラム本体>
     ....
 }
```