ACM ICPC 2016 国内予選 問題 A 被験者の選定

http://icpc.iisf.or.jp/past-icpc/domestic2016/problems/all_ja.html#section_A

- 全ての学生ペアの成績の差の絶対値の最小値を求めれば良い。即ち、学生 i の成績を

```
for (i=0; i<n－1; i++)
   for (j=i+1; j<n; j++)
      if(|a[i]－a[j]|<min) min = |a[i]－a[j]|
```

で求めることが出来る。n 人のペアは $\frac{n\times(n-1)}{2}$ 通りなので $O(n^2)$であるが、n <= 1000 なので問題無い。（pa1.c）

- 最初に成績でソートしておけば、隣通しの比較で最小値を求めることが出来る。即ち、学生の成績 a[i]が昇順にソート済みであるとすると

```
for (i=0; i<n－1; i++)
   if(a[i+1]－a[i]<min) min = a[i+1]－a[i];
```

で求めれば良い。平均の時間複雑度は、ソートに O(n log n)、隣同士の比較に O(n)なので全体として O(n log n)となる。（pa2.c）

【pa1.c】

```
/************************************************************
ICPC2016 Domestic Problem A 被験者の選定
http://icpc.iisf.or.jp/past-icpc/domestic2016/problems/all_ja.html#section_A
Filename:       pa1.c
Compile:        cc –Wall pa1.c
Check:          ./a.out < A0 > A0.result; diff A0.result A0.ans
Algorithm:      すべてのペアの成績の差を生成しながら最小値を求める O(n^2)
                学生数の最大値は 1000 なので、O(n^2)でも問題無い
************************************************************/

#include <stdio.h>
#define MAXN 1000                       // 学生の人数の最大値
#define MAXA 1000000                    // 成績の最大値
#define DIFF(x,y) ((x)>(y))?((x)-(y)):((y)-(x))         // 差の絶対値
int n;                                  // 学生の人数
int a[MAXN];                            // 学生の成績

int main(void)
{
    while(1){
        int i,j;
        int diff;                       // 成績の差
        int diff_min = MAXA;            // 成績の差の最小値
        scanf("%d", &n);                // 学生の人数の入力
        if(n == 0) break;               // n == 0 なら終了
        for(i=0; i<n; i++)              // 各学生の成績を入力
            scanf("%d", &a[i]);
        diff_min = MAXA;                // 成績の差は MAXA 以下
        for(i=0; i<n-1; i++){
            for(j=i+1; j<n; j++){
                diff = DIFF(a[i],a[j]);     // 学生 i と学生 j の成績の差
                if(diff < diff_min) diff_min = diff;   // 最小値の更新
            }
        }
        printf("%d¥n", diff_min);       // 答えのプリント
    }
    return 0;
}
```

【pa2.c】

```
/*************************************************************
ICPC2016 Domestic Problem A 被験者の選定
http://icpc.iisf.or.jp/past-icpc/domestic2016/problems/all_ja.html#section_A
Filename:       pa2.c
Compile:        cc –Wall pa2.c
Check:          ./a.out < A0 > A0.result; diff A0.result A0.ans
Algorithm:      まず成績順にソート(O(n log n))してから、
                隣通しの学生の成績の差の最小値を求める(O(n))
*************************************************************/

#include <stdio.h>
#include <stdlib.h>
#define MAXN 1000           // 学生の人数の最大値
#define MAXA 1000000        // 成績の最大値

int n;                      // 学生の人数
int a[MAXN];                // 学生の成績

// クイックソートのための比較関数
int compare(const void *p1, const void *p2)
{
    int ret;                // 戻り値
    int n1 = *(int *)p1;
    int n2 = *(int *)p2;
    if(n1 < n2) ret = -1;
    else if(n1 == n2) ret = 0;
    else ret = 1;
    return ret;
}

int main(void)
{
    while(1){
        int i;
        int diff;                       // 成績の差
        int diff_min = MAXA;            // 成績の差の最小値
        scanf("%d", &n);                // 学生の人数の入力
        if(n == 0) break;               // n == 0 なら終了
        for(i=0; i<n; i++)              // 各学生の成績を入力
            scanf("%d", &a[i]);
        qsort((void *)a, n, sizeof(int), compare);    // 成績をソート
        diff_min = MAXA;                // 成績の差は MAXA 以下
        for(i=0; i<n-1; i++){
            diff = a[i+1] - a[i];       // 学生 i と学生 i+1 の成績の差
            if(diff < diff_min) diff_min = diff;      // 最小値の更新
        }
        printf("%d¥n", diff_min);       // 答えのプリント
    }
    return 0;
}
```