

```

1  /*
2  给你个整数数组 arr，其中每个元素都 不相同。
3  请你找到所有具有最小绝对差的元素对，并且按升序的顺序返回。
4
5  示例 1:
6
7  输入: arr = [4,2,1,3]
8  输出: [[1,2],[2,3],[3,4]]
9
10 示例 2:
11
12 输入: arr = [1,3,6,10,15]
13 输出: [[1,3]]
14
15 示例 3:
16 输入: arr = [3,8,-10,23,19,-4,-14,27]
17 输出: [[-14,-10],[19,23],[23,27]]
18
19 提示:
20     2 <= arr.length <= 10^5
21     -10^6 <= arr[i] <= 10^6
22 来源：力扣（LeetCode）
23 链接：https://leetcode-cn.com/problems/minimum-absolute-difference
24 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
25 */

```

分析:

- 排序
- 第一次遍历求最小
- 第二次遍历找出最小的位置封装用于返回
- 返回

方法一:C++_排序

```

1  class Solution
2  {
3      public:
4          vector<vector<int>> minimumAbsDifference(vector<int>& arr)
5          {
6
7
8              vector<vector<int>> vvi                                     ;
9              int min_abs_diff = 2147483647 ;
10             int i = 0 ;
11
12             sort(arr.begin(),arr.end());
13             for( i = 1; i < arr.size() ; i++)
14             {

```

```
15         if(arr[i]-arr[i-1] < min_abs_diff)
16         {
17             min_abs_diff = abs(arr[i]-arr[i-1]);
18         }
19     }
20
21     for( i = 1 ; i < arr.size() ; i++)
22     {
23         if(arr[i]-arr[i-1] == min_abs_diff)
24         {
25             vector<int> vi;
26             vi.push_back(arr[i-1]);
27             vi.push_back(arr[i]);
28             vvi.push_back(vi);
29         }
30     }
31
32     return vvi;
33 }
34 };
35
36 /*
37 执行结果:
38 通过
39 显示详情
40 执行用时 :172 ms, 在所有 cpp 提交中击败了19.70% 的用户
41 内存消耗 :17.9 MB, 在所有 cpp 提交中击败了100.00%的用户
42 */
```