Find three closest elements from given three sorted arrays

> 给定三个数组A，B，C，分别在A，B，C中找到一个数A[i]、B[j]、C[k]，使得abs(A[i] -B[j])+abs(B[j] - C[k]) + abs(C[k] - A[i])最小

首先定义i，j，k三个变量分别遍历A、B、C三个数组，在遍历过程中：

用 minOfThree记录A[i]、B[j]、C[k]中最小的数，用maxOfThree记录A[i]、B[j]、C[k]中最大的数，diff表示maxOfThree与minOfThree的差值，只要不断的减小diff的差值，就达到了使得A[i]、B[j]、C[k]不断接近的目的

且在遍历过程中当diff = 0时，此时已经不可能找到比当前的A[i]、B[j]、C[k]更优的解了，直接跳出循环，否则：

如果A[i]是最小的，则i++，若B[i]是最小的，则j++，除了这两者之外，k++

```
#include <bits/stdc++.h>

using namespace std;

typedef vector<int> vi;

void closestEle(vi &A, vi &B, vi &C, int &N, int &M, int &K) {
    int diff = INT_MAX;
    int resI = 0, resJ = 0, resK = 0;
    int i = 0, j = 0, k = 0;
    while (i < N && j < M && k < K) {
        int minOfThree = min(min(A[i], B[j]), C[k]);
        int maxOfThree = max(max(A[i], B[j]), C[k]);
        if (maxOfThree - minOfThree < diff) {
            resI = i, resJ = j, resK = k;
            diff = maxOfThree - minOfThree;
        }
        if (!diff) break;
        if (A[i] == minOfThree) ++i;
        else if (B[j] == minOfThree) ++j;
        else ++k;
    }
    printf("%d %d %d\n", A[resI], B[resJ], C[resK]);
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N, M, K;
        scanf("%d %d %d", &N, &M, &K);
        vi A(N), B(M), C(K);
        for (int i = 0; i < N; ++i) scanf("%d", &A[i]);
        for (int i = 0; i < M; ++i) scanf("%d", &B[i]);
        for (int i = 0; i < K; ++i) scanf("%d", &C[i]);
        closestEle(A, B, C, N, M, K);
    }
}
```