# 题目

给定一个排序好的数组 arr ，两个整数 k 和 x ，从数组中找到最靠近 x（两数之差最小）的 k 个数。返回的结果必须要是按升序排好的。

整数 a 比整数 b 更接近 x 需要满足：

|a - x| < |b - x| 或者

|a - x| == |b - x| 且 a < b

示例 1：

输入：arr = [1,2,3,4,5], k = 4, x = 3

输出：[1,2,3,4]

示例 2：

输入：arr = [1,2,3,4,5], k = 4, x = -1

输出：[1,2,3,4]

提示：

1 <= k <= arr.length

1 <= arr.length <= 104

数组里的每个元素与 x 的绝对值不超过 104

# 分析

## 方法一：滑动窗口

// Sliding window

class Solution1 {

public:

vector<int> findClosestElements(vector<int>& arr, int k, int x) {

int minDelta = INT\_MAX;

int delta = 0;

int startIdx = 0;

for (int i = 0; i < k; i++) {

delta += abs(arr[i] - x);

}

minDelta = min(delta, minDelta);

for (int i = k; i < arr.size(); i++) {

delta -= abs(arr[i - k] - x);

delta += abs(arr[i] - x);

if (delta < minDelta) {

minDelta = delta;

startIdx = i - k + 1;

}

}

return vector<int>(arr.begin() + startIdx, arr.begin() + startIdx + k);

}

};

## 方法二：删除法

// Delete the the farther one of leftmost side and rightmost side element until k left

class Solution2 {

public:

vector<int> findClosestElements(vector<int>& arr, int k, int x) {

auto res = arr;

while (res.size() > k) {

if (x - res.front() > res.back() - x) {

res.erase(res.begin());

} else {

res.pop\_back();

}

}

return res;

}

};

class Solution3 {

public:

vector<int> findClosestElements(vector<int>& arr, int k, int x) {

list<int> nums(arr.begin(), arr.end());

while (nums.size() > k) {

if (abs(nums.front() - x) > abs(nums.back() - x)) {

nums.pop\_front();

} else {

nums.pop\_back();

}

}

return vector<int>(nums.begin(), nums.end());

}

};

## 方法三：二分查找

左闭右开区间写法：

class Solution4 {

public:

vector<int> findClosestElements(vector<int>& arr, int k, int x) {

int left = 0;

int right = arr.size() - k;

while (left < right) {

int mid = left + (right - left) / 2;

if (x - arr[mid] > arr[mid + k] - x) {

left = mid + 1;

} else {

right = mid;

}

}

return vector<int>(arr.begin() + left, arr.begin() + left + k);

}

};

闭区间写法：

class Solution5 {

public:

vector<int> findClosestElements(vector<int>& arr, int k, int x) {

int left = 0;

int right = arr.size() - k - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (x - arr[mid] > arr[mid + k] - x) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return vector<int>(arr.begin() + left, arr.begin() + left + k);

}

};