|  |
| --- |
|  |
| #include<vector> |
|  | using namespace std; |
|  |  |
|  | int getPivot(vector<int>& arr, int n) { |
|  |  |
|  | int s = 0; |
|  | int e = n-1; |
|  | int mid = s + (e-s)/2; |
|  |  |
|  | while(s<e) { |
|  |  |
|  | if(arr[mid] >= arr[0]) |
|  | { |
|  | s = mid+1; |
|  | } |
|  | else{ |
|  | e = mid; |
|  | } |
|  | mid = s + (e-s)/2; |
|  | } |
|  | return s; |
|  | } |
|  |  |
|  | int binarySearch(vector<int>& arr, int s, int e, int key) { |
|  |  |
|  | int start = s; |
|  | int end = e; |
|  |  |
|  | int mid = start + (end-start)/2; |
|  |  |
|  | while(start <= end) { |
|  |  |
|  | if(arr[mid] == key) { |
|  | return mid; |
|  | } |
|  |  |
|  | //go to right wala part |
|  | if(key > arr[mid]) { |
|  | start = mid + 1; |
|  | } |
|  | else{ //key < arr[mid] |
|  | end = mid - 1; |
|  | } |
|  |  |
|  | mid = start + (end-start)/2; |
|  | } |
|  |  |
|  | return -1; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | int findPosition(vector<int>& arr, int n, int k) |
|  | { |
|  | int pivot = getPivot(arr, n); |
|  | if( k >= arr[pivot] && k <= arr[n-1]) |
|  | {//BS on second line |
|  | return binarySearch(arr, pivot, n-1, k); |
|  | } |
|  | else |
|  | {//BS on first line |
|  | return binarySearch(arr, 0, pivot - 1, k); |
|  | } |
|  |  |
|  | } |
|  |  |