Lecture-14

Binary Search Interview Questions [Google, Amazon, Microsoft] || ProblemSet - 2

Q1. Find Pivot Element --> Element around which array is rotated

```
#include<iostream>
using namespace std;
int FindPivot(int arr[], int n){
int start = 0, end = n-1;
int mid = start + (end-start)/2;
while(start<end){
if(arr[mid]>=arr[0]){
start = mid + 1;
}
else{
end = mid;
}
mid = start + (end-start)/2;
}
return start;
}
int main(){
int arr[5] = {5, 6, 7, 1, 2};
cout<<"Pivot element is at index: "<<FindPivot(arr, 5);
return 0;
}
```

Q2. **Search In Rotated Sorted Array**

```
int getPivot(vector<int>& arr, int n){
int start = 0, end = n-1;
int mid = start + (end - start)/2;
while(start<end){
if(arr[mid]>=arr[0]){
start = mid+1;
}else{
end = mid;
}
mid = start + (end-start)/2;
}
return start;
}
int binarySearch(vector<int>& arr, int s, int e, int key){
int mid = s+(e-s)/2;
while(s<=e){
if(arr[mid] == key){
return mid;
}
else if(arr[mid]>key){
e = mid-1;
} else {
s = mid + 1;
}
mid = s + (e - s) / 2;
}
return -1;
}
int search(vector<int>& arr, int n, int k)
{
// Write your code here.
// Return the position of K in ARR else return -1.
int pivot = getPivot(arr, n);
if(k>=arr[pivot] && k<=arr[n-1]){
return binarySearch(arr,pivot, n-1,k );
}else{
return binarySearch(arr, 0, pivot-1, k);
}
}
```

Q3. Square root using binary search

```
int binarySearch(int n){
int s=0, e=n;
long long int mid = s+(e-s)/2;
long long int ans=-1;
while(s<=e){
long long int square = mid*mid;
if(square==n){
return mid;
}
if(square<n){
ans=mid;
s=mid+1;
}else{
e=mid-1;
}
mid = s+(e-s)/2;
}
return ans;
}
int floorSqrt(int n)
{
// Write your code here.
return binarySearch(n);
}
```