SCAN ALGORITHM:

#include <bits/stdc++.h>

using namespace std;

int size = 8;

int disk\_size = 200;

void SCAN(int arr[], int head, string direction)

{

int seek\_count = 0;

int distance, cur\_track;

vector<int> left, right;

vector<int> seek\_sequence;

// appending end values

// which has to be visited

// before reversing the direction

if (direction == "left")

left.push\_back(0);

else if (direction == "right")

right.push\_back(disk\_size - 1);

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

if (arr[i] < head)

left.push\_back(arr[i]);

if (arr[i] > head)

right.push\_back(arr[i]);

}

// sorting left and right vectors

std::sort(left.begin(), left.end());

std::sort(right.begin(), right.end());

// run the while loop two times.

// one by one scanning right

// and left of the head

int run = 2;

while (run--) {

if (direction == "left") {

for (int i = left.size() - 1; i >= 0; i--) {

cur\_track = left[i];

// appending current track to seek sequence

seek\_sequence.push\_back(cur\_track);

// calculate absolute distance

distance = abs(cur\_track - head);

// increase the total count

seek\_count += distance;

// accessed track is now the new head

head = cur\_track;

}

direction = "right";

}

else if (direction == "right") {

for (int i = 0; i < right.size(); i++) {

cur\_track = right[i];

// appending current track to seek sequence

seek\_sequence.push\_back(cur\_track);

// calculate absolute distance

distance = abs(cur\_track - head);

// increase the total count

seek\_count += distance;

// accessed track is now new head

head = cur\_track;

}

direction = "left";

}

}

cout << "Total number of seek operations = "

<< seek\_count << endl;

cout << "Seek Sequence is" << endl;

for (int i = 0; i < seek\_sequence.size(); i++) {

cout << seek\_sequence[i] << endl;

}

}

// Driver code

int main()

{

// request array

int arr[size] = { 176, 79, 34, 60,

92, 11, 41, 114 };

int head = 50;

string direction = "left";

SCAN(arr, head, direction);

return 0;

}

OUTPUT:

Total number of seek operations = 226

Seek Sequence is

41

34

11

0

60

79

92

114

176

--------------------------------

Process exited after 0.1141 seconds with return value 0

Press any key to continue . . .