1-Lower&Upper.Bound:

Binery search on index btw

Lower-bound(a,a+n,x); \*

يرجع مؤشر على x او على اول عنصر اكبر منه

لايرجاع مؤشر

Int p= lower-bound(a,a+n,x)-a;

لايرجاع قيمة

Int p= \* lower-bound(a,a+n,x);

Upper-bound(a,a+n,x); \*

يرجع مؤشر على اول اكبر عنصر من x

لايرجاع مؤشر

Int p= upper-bound(a,a+n,x)-a;

لايرجاع قيمة

Int p= \* upper-bound(a,a+n,x);

Note::

لايرجاع مؤشر على اصغر تماما من x

Int p= lower-bound(a,a+n,x)-a-1;

لايرجاع مؤشر على اصغر او يساوي من x

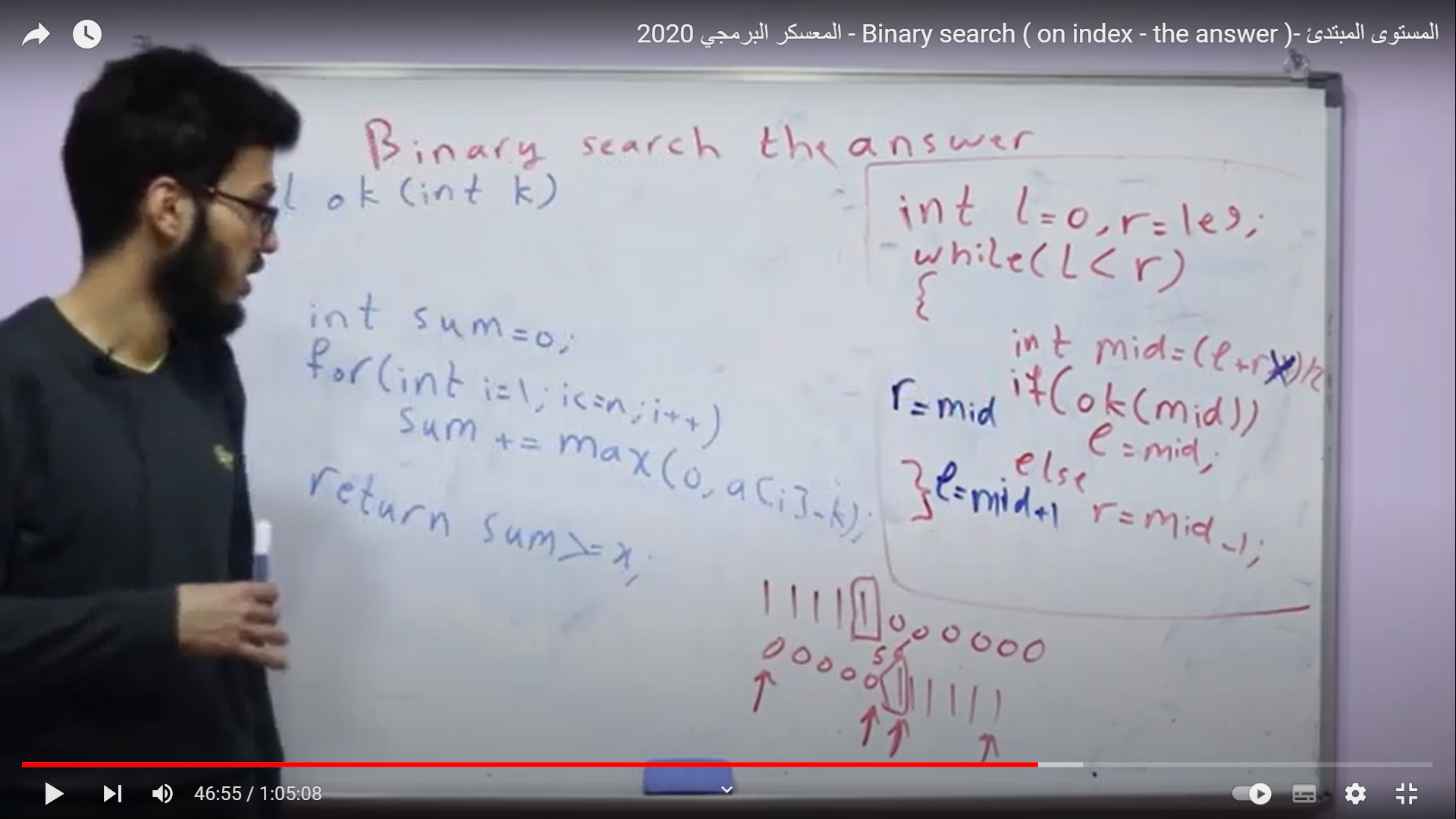
Int p= upper-bound(a,a+n,x)-a-1;

ملاحظة : للست بس منحط s.lower or upper و للماب منحط map

2-Binery seach:

Binery search on answer btw

1. #include <bits/stdc++.h>
2. using namespace std;
4. #define ll long long
5. #define F first
6. #define S second
7. #define ii pair < int , int >
8. #define ever (;;)
10. bool ok(int mid)
11. {
13. }
15. int main()
16. {
17. /// 00000011111
19. int low = 1 , high = 1e9;
20. while( low < high )
21. {
22. int mid = (low+high)/2;
24. if( ok(mid) )
25. high = mid;
26. else
27. low = mid+1;
28. }
30. cout << low;
32. /// 1111110000000
34. int low2 = 1 , high2 = 1e9;
35. while( low2 < high2 )
36. {
37. int mid = (low2+high2+1)/2;
39. if( ok(mid) )
40. low = mid;
41. else
42. high = mid-1;
43. }
45. cout << low2;
46. }



####################################################

Frequnce array

Int size=x;

Int rangeSize=y;

Int values[size]={a,b,b,c,c,…};

Int freqArray [rangeSize]={a,b,c,…}

For(int i=0;i<size;i++)

{

freqArray[values[i]]++;

}

For(int i=0;i<rangeSize;i++)

Cout<<i<<”\_”<<freqArray[i]<<endl;

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

Deviders of a number big O of squer

cin>>n;

for(int i=1;i\*i<=n;i++){

if(n%i==0)

cout<<i<<endl;

if(n/i!=i)

cout<<n/i<<endl;

}

||||||||||||||||||||||||||||||||||||||||||||||||||||||||

// CPP program to count frequencies of array items

#include <bits/stdc++.h>

using namespace std;

void countFreq(int arr[], int n)

{

// Mark all array elements as not visited

vector<bool> visited(n, false);

// Traverse through array elements and

// count frequencies

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

// Skip this element if already processed

if (visited[i] == true)

continue;

// Count frequency

int count = 1;

for (int j = i + 1; j < n; j++) {

if (arr[i] == arr[j]) {

visited[j] = true;

count++;

}

}

cout << arr[i] << " " << count << endl;

}

}

int main()

{

int arr[] = { 10, 20, 20, 10, 10, 20, 5, 20 };

int n = sizeof(arr) / sizeof(arr[0]);

countFreq(arr, n);

return 0;

}

<p style="width: 50%; margin: auto;" class="bab">I played Basketball since 2015 and I have been a national team player since 2018</p>