Q1. Determining the end of an array.

A. 🡪

Algorithm:

|  |
| --- |
| 1. First find value Start checking the elements at index two to the power ‘i’, that is 2, 4, 8, 16, 32, 64, …, 2i and stop once you have found the first occurrence of ‘Infinity’. Suppose you stop at an index ‘k’ 2. Now use divide and conquer technique to find out the index of last number that is not infinity.   *if(array[array.length] != INF)*  *return array.length;*    *k=0*  *element = array[0]*  *while(element != INF)*  *k = k + 1*  *element = array[2k]* …STEP1    *start=[2k-1]*  *end =[2k]*    *while(start < end)*  *mid=(start + end)/2*  *if(array[mid]! = INF)*  *start = mid + 1*  *else*  *end = mid - 1*    *return (end - 1);* …STEP2 |

Complexity:

|  |
| --- |
| 1. As we are moving in the range of power of 2, and there are total ‘n’ non-infinity elements, first step of algorithm would take log(n) + 1 elements, as in the worst case to get first infinity we might travel to ‘2n’ index. Therefore, running time would be log(2n) = log2 + log(n) = log(n) + 1 2. In second step as we do a divide and conquer search between *2k-1* to *2k* elements, the running time would be log(n) 3. Thus, total complexity would be   f(n) = log(n) + 1 + log(n)  = 2\*log(n) + 1  f(n) = O(log(n)) … [as log(n/2) < log(n)] |

Q2(a) **Part 1** k-colorable graph with ∆.

A. 🡪

General Proof –

When we say, a graph has a degree ∆, we mean that vertices in graph has maximum ∆ edges coming out from it. Thus, we can say that there are ∆ neighboring vertices for a vertex. Hence, including the starting vertex there are total ∆ + 1 vertices. Now suppose all these vertices are connected to each other, then we will require maximum ∆ distinct colors to color whole graph.

Proof by Induction –

**BASE CASE**

1. ∆ = **0**

colors required = ∆ +1 = 0 + 1 = **1**

This means the vertices in the graph are not connected to each other at all. Coloring all the vertex with same color, we stick to the constraint that no two neighboring vertices have same color.

1. ∆ = **1**

colors required = ∆ +1 = 1 + 1 = **2**

This means that every vertex in the graph is connected to exactly one other vertex. Coloring two connected vertices with two different color, we stick to the constraint that no two neighboring vertices have same color.

**INDUCTION STEP**

For value k such that, **0 < k < ∆**

1. We know that a graph G with degree ‘k’ can be colored in ‘k + 1’ distinct colors.
2. Now suppose we add one more vertex in the graph G, such that the new node is connected to all other vertices in the graph and let`s call new graph as G` (This is worst case, in the best case new vertex would be connected to only one other or no other vertex in G).

degree of new graph G` = 1 + degree of old graph G

= 1 + (k)

= k + 1

This means that degree of G` would be degree of G plus one, which is ‘k +1’.

Now, since new vertex is connected to all other vertices in graph G, will have to give a new color to the new vertex.

total colors required for G` = 1 + total colors required for G

= 1 + (k + 1)

= k + 2

We require k + 2 distinct colors to color graph G`

1. Hence, from induction step 1 and 2 we can say that, we require k + 2 colors to color graph which has degree k +1.

Q2(a) **Part 2** Construct graph with maximum degree ∆ and that is ∆ + 1 colorable.

A. 🡪

Basic Logic –

Construct a *strongly connected graph* containing ∆ + 1 vertices. A graph is strongly connected if each vertex is connected to all every other vertex in graph. And we require ∆ edges coming out from each vertex to connect it to every other index in graph, thereby making the degree of graph as ‘∆’.

Please refer following image for further understanding.

