Algorithm Project - Answers

Name: Aimee
Ramirez

Algorithm 1 : Find the Maximum 8

1. Instructions for the Algorithm

In "find the maximum", Player is to find the largest number in the random list of numbers. The result is a definitive maximum element out of the list of random numbers. Everything in * is considered a rule to be thought of as the game goes on.

. There is a start and end, I'll call it "Game". This Game can start or end. This is like a switch or tiny list to be checked in start/end or on/off. If start, we go to the first thing, if not we terminate the game.

Game = [start, end] First thing I would do is create my random set of numbers to be shown in a list. The way I would do this is to have two lists of integers from [0 through 9]. It is having two lists within a list. I'll call the two lists to be A and B.

A = [0,1,2,3,4,5,6,7,8,9] B = [0,1,2,3,4,5,6,7,8,9] This list can only hold the two, A and B, lists to form the one list to be created. This one list will be called the "Number" list.

n = Number = [A,B] For every instance of A, only one of the integers can be chosen. The same goes for B in forming the Number list which only holds two values. If the values come up as [0,0], this is to be excluded for representing the value structure.

Number = [A,B] can not be [0,0] Number can be [0,3] or [9,9] or [2,4] and so on to represent the values of A and B. Number is to be set to a whole value for A is set to tens place as B is ones place of the Number list.

The examples given above was [0,3], this will set to 03 which is equal to 3. If to be [9,9] is to be set as 99.

We know that this Number list can only have: *The maximum element of to 99. * The lowest to be 01. Since we excluded 0/00/[0,0] as an option, we are going to iterate over these numbers to be created to Number list.

There will be another list to be created, this will be called Container. This Container is to hold Number list. This Container can hold many Number list but Number can not be repeated in the Container. This Container can have 8 of Number, 15 of Number, and 25 of Number as items in this list option. I'll represent the Container to be [ ] like a box with items of n/Number.

Algorithm Project - Answers

Container = [8 options of Number = Container[1], 15 options of Number = Container[2], 25 option of number = Container[3]]

[A] and [B] can be the same number [A,B] such as [5,5]. [B] less than or greater than [A] this is valid. [A] is greater or less than [B] this is valid. [5,9] to [59] but [9,5] to [95] can be valued as a set.

*Number list has 01-99 if 0 is excluded. *The minimum of Number list to be held in the Container is 8. *The maximum of Number list to be held in the Container is 25.

Groups of Number of [A,B] can not be repeated in Container list once it is set!

For each time Player enters the room/ starts the game, Container is to be adjusted to give the Number list to Container of choice for the game.

If Player, ends the Game or sets restart, this is to start the Game from the first step and onward.

When the Game starts, the Player can choose from left being Number[1] to the right of the Container list being the max of either Number[8], Number[15], or Number[25].

The Player can toggle the items in Container but this is to be done once to be checked the value in Number list set of elements of [A,B]. If selected/toggled on item, this will pull the value to show the Number with [A,B] to be seen/calculated. If Number set [A,B] is to be evaluated, the next chosen item selected is to be compared to.

If [A] is to be seen knowing [A] is in the tens place. This can compare all the other tens place if to be less than or greater than it's given value. The other option is we can either evaluate it as a whole number of [A,B] as the given value.

For Example: Key: "...":
"and so on or etc."

Visualization: Container =[
Number[1] = [5,9], ...] →
Container = [[[5],[9]] or 59], ...]

Two ways to check as max value: So far we have A and B lists to belong to Number list that only Number list holds 2 lists in it. Many of Number list belong to Container with the minimum of this box to hold 8 and maximum is 25 Number list of groups in 2.

Algorithm Project - Answers

[A,B] belongs to Number →
Number belongs to Container →
Container has Many of Number
list

When the Container is set, it is time to compare the values/elements to each item/Number list set to be pulled.

1. a) [5,9] we can pull [5] as a given element. When toggling the next item of Number set in Container, we get
[9,5].

We can pull the [A] as [9]. This we can quickly compare that [5] < [9] as an element to be registered. So without having to check the B value, we can say [9,5] or [95] is the maximum value selected. b) If in the tens place of the first [A]/[5] is the same/exact element the

next item of second [A]/[5] then we can evaluate if the first [B]/[5] is greater or lesser value to the second [B]/[9] since there is no repeats of the groups of Number in Container list.

First[5,5] is less than
Second[5,9]

O
R

2. We can evaluate, [5,9] to be joined as [59] and [9,5] to be joined as [95].

First[59] is less than
Second[95].

This we can determine if true it is more than the other if not to be false to continue the steps until max value is reached. *This is considering unoptimized algorithm.* Game = start

1. Container[1] = [Number[1],Number[2], Number[3],Number[4], Number[5],Number[6], Number[7],Number[8]]

Problem
Size:8

Container = [[0,1],[0,2], [0,3], [0,4],
[0,5],[0,6],[0,7],[0,8]]

If [A] is all [0], then check [B] item
value. If 1[B] = 1 , 2[B] = 2 then 1< 2
Or 1...7 < 8 So **8** is the maximum value
given.

2. Number of Steps for Problem

Size: 15

Game =
start

Algorithm Project - Answers

2. Container[2] = [Number[1],Number[2], Number[3],Number[4], Number[5],Number[6], Number[7],Number[8], Number[9],Number[10], Number[11],Number[12], Number[13],Number[14], Number[15]]

Problem Size:15 Container = [... < [1,1],[2,2], [2,3], [3,4], [4,5],[5,6],[6,7],[7,8]] If 1[A] is [1], 2[A] is 2. Then 2[A,B] moves forward in validation. If 2[A] is equal to 3[A]. then check [B] item value. [2] < [3]. 3[A,B] move forward. As well we can compare next pairs of A's as 2>3 → 3>4 →4>5 →5>6 →6>7→7 If [7,8] with the value of **78** is the maximum value given. Game = end

3. Number of Steps for Problem Size: 25__

Game = start i = iteration of a list

n = selected. ... = is parameter of a list

3. Container[3] = [Number[1],Number[2], Number[3],Number[4], Number[5],Number[6], Number[7],Number[8], Number[9],Number[10], Number[11],Number[12], Number[13],Number[14], Number[15], Number[16],Number[17], Number[18],Number[19], Number[20],Number[21], Number[22],Number[23], Number[24],Number[25]] If Container= [[...Number[1 through 25 items has different amount for each value[i]]] and if these options < n that is [97]]

97 is the maximum value given.

Game =
end

4. Formula for the Number of Steps (S) **S = {Comparison equal to minimum Moves/Steps taken to find the max value which is the total of Number list in Container.}** Summary = {

i= iteration instance ++ = can increase/unlimited ... = is parameter of a list

Algorithm Project - Answers

Number = item Value
= amount of sum Max
= maximum n =
selected

- Item/Number can be n but n is the selected item to compare.
- There is many of Number in Container list that the items in Container list can be unlimited.
- That Number belongs to Container list if so Number value can be unlimited.

Container =[... Number [[A],[B], ...[i]++]++] Or Container = [... Number[1...i++]++]

Or Container = [...Number[value = 1 through value = unlimited] items can be

unlimited]

if i is not [0,0] or 0 in value.{ then value++ if value is an item as Number in Container. And different value holds of item as Number list of each in

Container. Then evaluate the value between selected item/n. } This item/n is

then compare selected item/n to the second selected item’s value. If item/n is more than the second value item , Then if true, n will be the max value between the first and second item. If item/n is less than the second value item, Then it’s false, keep the max value selected which is the second item in this case.

And remove the lowest value item in the array from selection to be triggered once in moves of validation. Move on to the next item and compare kept in the last max value to the new value item. Then repeat above for true or false comparisons until reaching the end.

} }