SEARCHING

It refers to the operation of finding the location of any item in the array. The search is said to be successful if the item is found; otherwise it is unsuccessful.

Linear search

Compare the given item i.e. to be searched with each element of the array one by one. This method traverses the data sequentially is called linear or sequential search

Linear_Search.c

Algorithm for Linear Search

A → a linear array with size N
N → size of the array
ITEM → item to be searched
LOC → location of the element
FOUND → Boolean variables

STEP 1: [Initialization]
FOUND := FALSE; K:=0
STEP 2: [Read the value i.e to be searched]
Read ITEM
STEP 3: [Comparing each element with the ITEM]
Repeat while K<N
If A[K] = ITEM
FOUND := TRUE
GO TO STEP 4
Otherwise
K := K+1

STEP 4: [Checking for the value of FOUND]
If FOUND = TRUE
Write "Record Found"
Otherwise
Write "Record Not Found"

STEP 5: [FINISHED]
EXIT

2018/8/31 15:39

Binary Search

Example of searching a name in the dictionary.

In binary search algo, during each stage of our algorithm, our search for an ITEM is reduced to a segment of elements of DATA.

DATA[BEG], DATA[BEG+1], - - - DATA[END]

BED & END are beguing & end locations of segment

→ The algo. compares ITEM with the middle element DATA[MID] of the segment. MID is obtained by

MID = INT((BEG+END)/2)

→ If DATA[MID] = ITEM search successful

→ If ITEM < DATA[MID]

Set END := MID-1 & begin searching

Else

BED = MID+1 & Begin searching.

Initially BEG=1 & END=N or BEG=LB & END=UB

Algorithm: BINARY(DATA, LB, UB, ITEM, LOC)

DATA → Sorted array with lower bound LB & Upper Bound UB.

LOC → location of ITEM found

initially LOC = NULL

2018/8/31 15:41

STEP 1 [Initialisation]
Set BEG := LB, END := UB & MID = INT ((BEG + END)/2)
2. Repeat steps 3 & 4 while BEG ≤ END & DATA[MID] ≠ ITEM
3. If ITEM < DATA[MID] then
Set END := MID - 1
Else
Set BEG := MID + 1
[End of If structure]
4. Set MID := INT ((BEG + END)/2)
[End of Step 2 loop]
5. If DATA[MID] = ITEM then
Set LOC := MID
Else
Set LOC := NULL
[End of If structure]
6. Exit

Example :
Given
11, 22, 30, 33, 40, 44, 55, 60, 66, 77, 80, 87, 88

Search ITEM = 40

1. Initially BEG = 1 & END = 13
MID = (1+13)/2 = 7 so DATA[MID] = 55
2. 40 < 55 END = MID - 1 = 6
MID = (1+6)/2 = 3 DATA[MID] = 30
3. Since 40 > 30, BEG = MID + 1 = 4
MID = (4+6)/2 = 5 so DATA[MID] = 5
Here LOC = MID = 5 Found.

(b) Search ITEM = 85
1. Again initially BEG = 1, END = 13, MID = 7
DATA[MID] = 55
2. 85 > 55 BEG = MID + 1 = 8, END = 13
MID = (8+13)/2 = 21/2 = 10 DATA[MID] = 77
3. 85 > 77 BEG = MID + 1 = 11
MID = (11+13)/2 = 12 so DATA[MID] = 88
4. 85 < 88 END = MID - 1 = 11
MID = (11+11)/2 = 11 DATA[MID] = 80
As BEG = END = MID
ITEM not found

2018/8/31 15:41