17-2 Making binary search dynamic

只需實作insert、search及 print

Binary search of a sorted array takes logarithmic search time, but the time to insert a new element is linear in the size of the array. We can improve the time for insertion by keeping several sorted arrays.

Specifically, suppose that we wish to support SEARCH and INSERT on a set of n elements. Let $k = [\lg(n + 1)]$, and let the binary representation of n be $\langle n_{k-1}, n_{k-2}, \ldots, n_0 \rangle$. We have k sorted arrays $A_0, A_1, \ldots, A_{k-1}$, where for $i = 0, 1, \ldots, k - 1$, the length of array A_i is 2^i. Each array is either full or empty, depending on whether $n_i = 1$ or $n_i = 0$, respectively. The total number of elements held in all k arrays is therefore $\sum_{i=0}^{k-1} n_i 2^i = n$. Although each individual array is sorted, elements in different arrays bear no particular relationship to each other.

a. Describe how to perform the SEARCH operation for this data structure. Analyze its worst-case running time.

b. Describe how to perform the INSERT operation. Analyze its worst-case and amortized running times.

參考輸出格式 :

```
Dynamic Binary Search Operation:
(1) Insert
(2) Search
(3) Delete
(4) Print
(5) Exit
Input a operation...>4
0
0 0
9 10 11 12
2 3 4 5 6 7 8 100
Press any key to continue . . .
Dynamic Binary Search Operation:
(1) Insert
(2) Search
(3) Delete
(4) Print
(5) Exit
Input a operation...>1

***Insert***
Input a number...>22
Press any key to continue . . .
Dynamic Binary Search Operation:
(1) Insert
(2) Search
(3) Delete
(4) Print
(5) Exit
Input a operation...>4
22
0 0
9 10 11 12
2 3 4 5 6 7 8 100
Press any key to continue . . .
```

```
Press any key to continue . . .
Dynamic Binary Search Operation:
(1) Insert
(2) Search
(3) Delete
(4) Print
(5) Exit
Input a operation...>2

***Search***
Input a number...>100
100: [3][7]
Press any key to continue . . .
Dynamic Binary Search Operation:
(1) Insert
(2) Search
(3) Delete
(4) Print
(5) Exit
Input a operation...>3
```