

THEORETICAL ASSIGNMENT № 2

Harsh Sinha(14265), Deepak Gangwar(14208)

09/03/2017

Problem 1

The pseudo code is as follows.

Listing 1: Pseudo code – Search in an Infinite Array.

```
1  BOOL infinite_search{
2  isfound = FALSE;
3  i = 1;
4  s = s;
5  while (isfound == 0)
6  {
7      if (A[i] == s)
8          isfound = TRUE;
9      else if (A[i] < s)
10         i = i * 2;
11     else if (A[i] > s OR A[i] == empty)
12         modif_bin_search (i);
13 }
14 return isfound;
15 }
```

Listing 2: modif_bin_search.

```
1  void modif_bin_search (int i)
2  {
3      L = i / 2;
4      R = i;
5      mid = (L + R) / 2
6      while (mid != R)
7      {
8          if (A[mid] == s)
9              return;
10         else if (A[mid] < s)
11             L = mid + 1;
12         else if (A[mid] > s OR A[mid] == empty)
13             R = mid -1;
14     }
15 }
```

Proof of correctness

TODO fill here.

Time Complexity

The function `modif_bin_search` takes $\log(i)$ steps to give us a solution (Similar to a binary search). Now the original function in the worst case takes $\log(n)$ steps. Thus in the worst possible case our algorithm shall consume $2 * \log(n)$. Hence, the order of the given algorithm is $\log(n)$.

This can also be proposed by saying that the number of elements that this algorithm skips keeps on doubling every loop and so the order should be $\log(n)$.

Notice that the base of $\log(n)$ is decided based on our choice of the multiplier in every step.

Problem 3

Part (a) Deleting 55

Part (b) Inserting 34



