Chandan C Bo
1BM18CS026

ADS

Skiplist

Insert

```
void skiplist :: insert (int key)
{
    Node * curr = head;
    Node * update [maxlvl+1];
    memset (update, 0, sizeof (Node*) * (maxlvl+1))

    for (int i = level ; i >= 0; i++)
    {
      while ( curr -> forward[i] != NULL &&
                  curr -> foward[i] -> key < key )
      {
          curr = curr -> forword [i] ;
      }
      update [i] = curr ;
    }
    curr = curr -> forward [0] ;
   if (curr == NULL || curr -> key != key )
   {
        int rlvl = random level ();
        if (rlvl > level )
        {
          for (int i = level+1 ; i < rlvl +1 ; i++ )
              update [i] = head ;
          level = rlvl ;
        }
       Node * newra = create Node (key, rlvl);
     for ( int i = 0 ; i <= rlvl ; i++ )
     {
        neon -> forward [i] = update [i] -> forward [i]
        update (i) -> forward (i) = n;
     }
}}
```

Chandan C Bagan
1BM18CS026

```
void Skiplist :: delete (int key)
{
    Node * curr = head ;
    Node * update [maxlevel + 1] ;
    memset ( update, 0, size of (Node *) * maxlevel+1)) ;
    for ( int i = level ; i >= 0 ; i-- )
    {
        while ( curr -> forward [i] != NULL &&
                        curr -> forward [i] -> key < key )
        {
            curr = curr -> forward [i] ;
        }
        update [i] = curr ;
    }
    curr = curr -> forward [0] ;
    if (curr != NULL && curr -> key == key)
    {
        for (int i = 0 ; i <= level ; i++ )
        {
            if (update[i] -> forward[i] != curr)
                break ;
            update [i] -> forward [i] = curr -> forward [i] ;
        }
    }
    while ( level > 0 && head -> forward [level] == 0)
    level-- ;
}
} ;
```

Chandan C Bagan
1BM18CS026.

```
void skiplist :: search ( int key ) {
    Node * curr = head ;
    for ( int i = level ; i >= 0 ; i -- )
   {
     while ( curr → forward [i] != NULL && curr → forward[i] →
                                                          < key )
         curr = curr → forward (i) ;
   }
   curr = curr → forward (0) ;
   if ( curr != NULL && curr → key == key )
      cout << " Found " << key << endl ;
};
```