```
Charden c ses
             ADS
Skiplist
Insout
void skipuist :: insent (int key)
   Node of curr = head;
   Node * updout [marb1+1]:
   memet ( update, o; size of ( node+) + ( markit)
   to 8 (int i = level; 1>=0; 1+1)
    while ( word + forward[ ] 1 = NULL yy
                 wer - pawara[i] - key < key)
       curr = curr - forwara [i]:
    update [] = curr :
  cour = cour - forward [0]:
 if ( aur == NULL | ) corr -> key 1 = key )
      int The = random level ();
      if [ rav 1 > level )
       for fint is lever+1; ix relot +1; i++)
           update [i] : head ?
       level = rlvl;
     Node & new = create Node ( text, tely !)
   for (int i=0; 1 <= r (v); i++)
      recon - forward [f] = update (;) -> forward!
  q update (i) - forward () = n?
```

```
chardon c Bagan
                                  1BM 180502 6
void stiplist :: delete ( int key)
  Node * curr = heard;
   Node * update [manievel +1]
   menset ( update, 0, size of ( Nade + ) + marxles 1+1));
   too (int i= level ; i >= 0; i -- )
      while I cover - forward (1) 1 = NULL & &
                 um - forward (i) - key < key)
         cour = cours -> forward [1]:
      update [1] = uvr;
      cure = cur - forward (0)
   of [ worr = HULL yg curr - key == key)
       for (distrize; ix=level; i++)
          4 (update[i] - forward[i] ! = aur)
          update [i] - forward [i] = wur - forward [i]
   Drue ( devel 20 y & nead - forword (level ) == 0)
    devel -- ;
```

IBMIS CS DZ void stiplist :: search ( int key ) [ Node \* cour = head = for (in i = level; i == 0; i --) where [ were forward [ ] ] = NULL &y wer - forward aver = aver -> forward (1): con = com - jourara (0); if ( aver 1 = NULL glo Cover -> key - = key ) coul K " Found " I' key K end !;