

```

int maxVal(treeNode* t, int k) {
// pre : t is a pointer to a binary tree, k is an integer;
// all items in t are non-negative
// post : returns the smallest item in t which is at least the value of k;
// returns -1 if there is no such item in t.
{if (t==NULL)
return -1;
else { int a= maxVal(t->left, k); // get the results for the subtrees
      int b= maxVal(t->right, k);
      if (a==-1 || b== -1) {a = a MAX b;} else { a= a MIN b;} // Combine them for the result

      if (a == -1) { if (t->item >= k) return t-> item; else return -1;}
      else (if t->item < k) return a; else return (t->item MIN a);
}
}

```

First get the values from the subtrees (a, b), then combine with the current value to get the overall result.

To combine the values (a, b, current), compute the smallest of the three which is at least k, but only out of those values which are at least k.

There are many possibilities for calculating this, any one of which would be correct.