PAGE NOW for Space   PAGE NOW for Joseph     PAGE NO	1 20	Kth largest sum in a Binary True
int height (struct theepode "noot)?  if (noot == NULL)  neturn 0;  else 8  int Theight = neight (noot > left);  int sheight = neight (noot > right);  if (sheight > vheight)  veturn sheight + 1;  else  veturn rheight + 1;  else  veturn rheight + 1;  ?  Void afs(struct Theepode * root, int level, long * sums) f  if (noot == NULL)  return;  sums[level] = sums[level] + root > NULL;  if (root > left)  afs (noot > left, level+ L, sums);  if (root > right)	Jue et	PAGE NOT 105/2024  DATE: 105/2024
else ?  int Theight: neight (root - left);  int sheight: neight (root - right);  if (sheight > rheight + 1;  else  veturn sheight + 1;  else  veturn rheight + 1;  ?  void afs(struct Treewode * root, in-1 level, long * sams) f  if (root = pull)  return;  sums[level] = sams[level] + root > pull;  if (root - left)  afs (root - left, levelt L, sams);  if (roof - right)		
elses  int theight: neight(root -> left);  int sheight: neight(root -> right);  if (sheight > rheight + 1;  else  veturn sheight + 1;  else  veturn rheight + 1;  q  void afs(struct Treepode + root, in-1 level, long + sams) f  if (root == NULL)  return;  sums[level] = sams[level] + root -> NULL;  if (root -> left)  afs (root -> left, level+L, sams);  if (roof -> right)		if (root == NULL)
int Theight = neight (root -> left);  int sheight = neight (root -> right);  if (I height > vheight + 1;  else  veturn theight + 1;  else  veturn vheight + 1;  ?  Void afs(struct Theepode of root, int level, long of Sams)?  if (root == pull)  return;  Sums[level] = Sums[level] + root -> NULL;  if (root -> left)  afs (root -> left, level+ L, sums);  if (roof -> right)		neturn o;
int sike gnt = neignt (noot > right);  if (I height > rheight + 1;  else  veturn theight + 1;  q  roid afs(struct Treepode * root, int level, long * Sams) f  if (noot == NULL)  return;  sums[level] = sums[level] + root > NULL;  if (root > left)  afs (root > left, level+ L, sums);  if (roof > right)		
it (sheight > vheight + 1;  veturn sheight + 1;  veturn rheight + 1;  void afs(struct Treewode root, int level, long sams) f  it (noot == NULL)  return;  sum of level] = sums[level] + root > NULL;  if (root > left)  afs (root > left, level+ L, sums);  it (roof > right)		int theight = neight (root → left);
return theight + 1;  else  veturn rheight + 1;   return rheight + 1;   void afs(struct Treepode * root, int level, long * sums) f  if (root == pull)  return;  Sums[level] = sums[level] + root > pull;  if (root > left)  afs (root > left, level+ 1, sums);  if (roof > right)		int sikeight = neight (noot > right);
return wheight +1;  you'd afs(struct Treewode froot, in-level, long sams)?  if (noot == NULL)  return;  Sums[level] = sums[level] + root > NULL;  if (root > left)  afs (root > left, level+ L, sums);		it (theight > vheight)
return Theight +1;		neturn kneight + 1;
you'd afs(struct Treepode * root, in-1 level, long * Sums) {  if (root == NULL)  return;  Sum & [level] = Sums[level] + root > NULL;  if (root > left)  afs (root > left, level+ L, Sums);  if (roof > rignt)		el 5-e
yo'd afs(struct Treewode * root, in-1 level, long * Sums) {  if (root == NULL)  return;  Sums[level] = Sums[level] + root > NULL;  if (root > left)  afs (root > left, level+ L, Sums);  if (roof > rignt)		return rheight +1;
void afs(struct Treewode * root, in-1 level, long * sums) {  if (root == NULL)  return;  sums[level] = sums[level] + root > NULL;  if (root > left)  afs (root > left, level+L, sums);  if (roof > rignt)		
void afs(struct Treepode * root, in-1 level, long * sums) {  if (root == NULL)  return;  Sum & [level] = sums[level] + root > NULL;  if (root > left)  afs (root > left, level+ L, sums);  if (roof > rignt)		1 (shub) > +
"+ (root == NULL)  return;  Sums[level] = Sums[level] + root > NULL;  if (root > left)  afs (root > left, level+ L, sums);  if (roof > rignt)		
return;  Sums[level] = Sums[level] + root > NULL;  if (root > left)  afs (root > left, level+ L, sums);  if (roof > rignt)		void afs (struct Treepode + root, int level, long + sums) &
Sums[level] = Sums[level] + root > NULL;  if (root > left)  afs (root > left, level+ L, Sums);  if (roof > rignt)		
if (root → left)  afs (root → left, level+ L, sums);  if (roof → rignt)		
afs (root > left, level+ L, sums);  it (root > rignt)		Sums[level]= sums[level] + root → NULL;
it (roof -> rignt)		17 (2001 - (EL))
it (roof -> right)		
l d and total d a		if (roof > right)
do S ( soot > right, level to, sums,		dfs (noot > right, level+1, sums);
9		9
3 2 1_4 m A 1		
long long kth lar yestlevelsum (struct mee No de not,		long long kth langest level sum (struct Thee No de nod,
int K)?		
int naheight (root);		int naneignt (root);
of(k)n)		if(k)n)
return-2;		
Tong rong & sums: (long long +) callo c(h,		rong rong & sums: (long long ) callo c(h,
3:3eof(conglong));		
ats (root, o, sums);		

