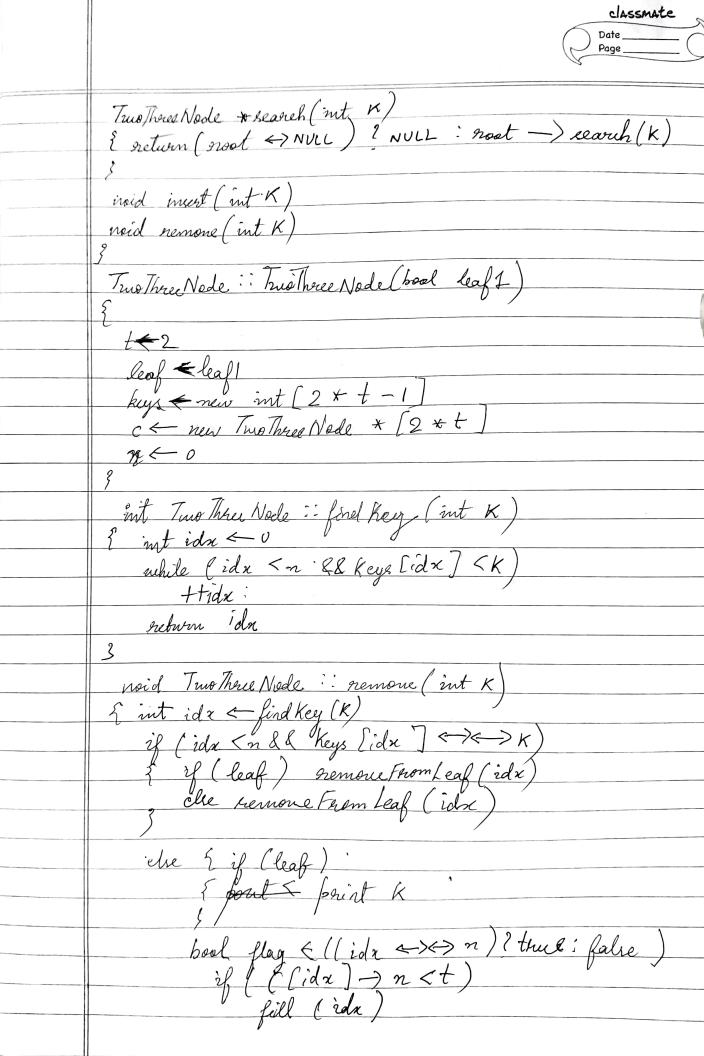
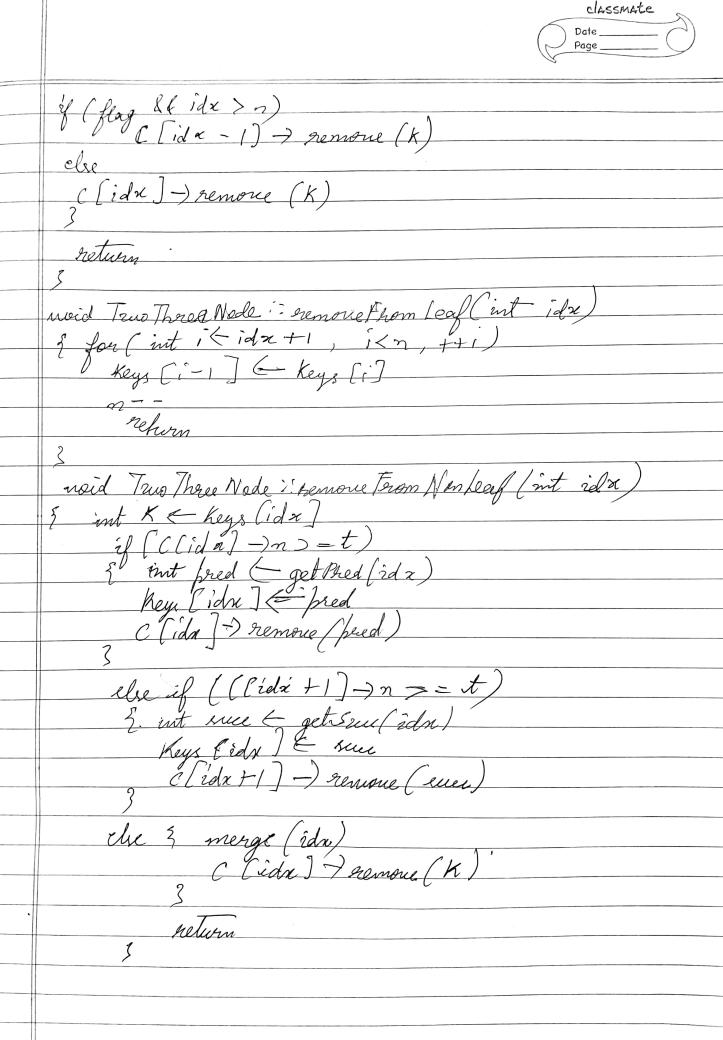
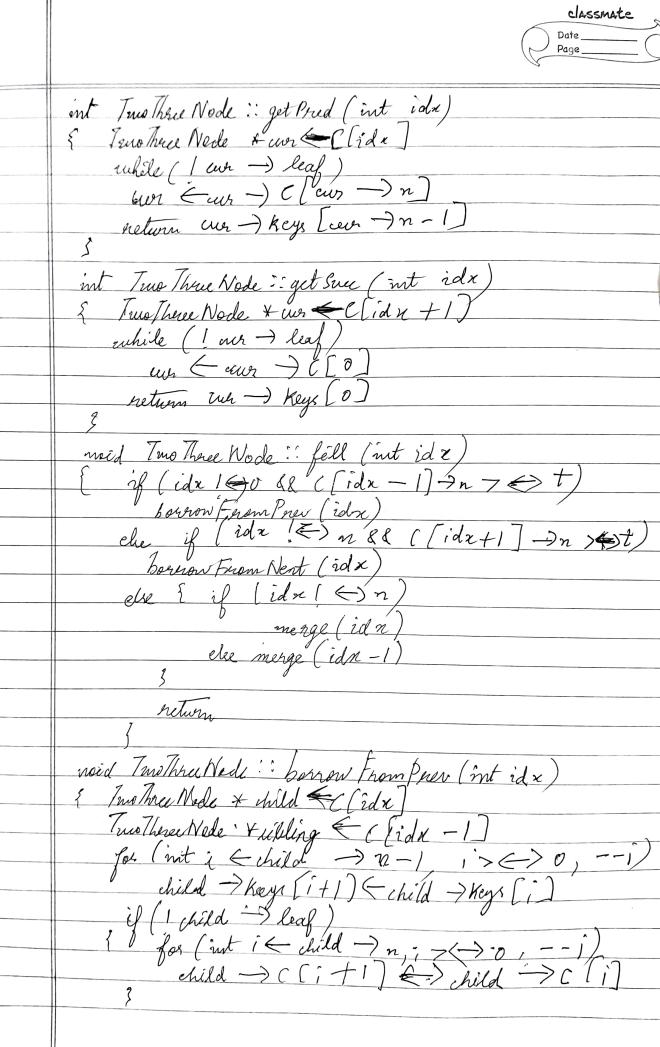
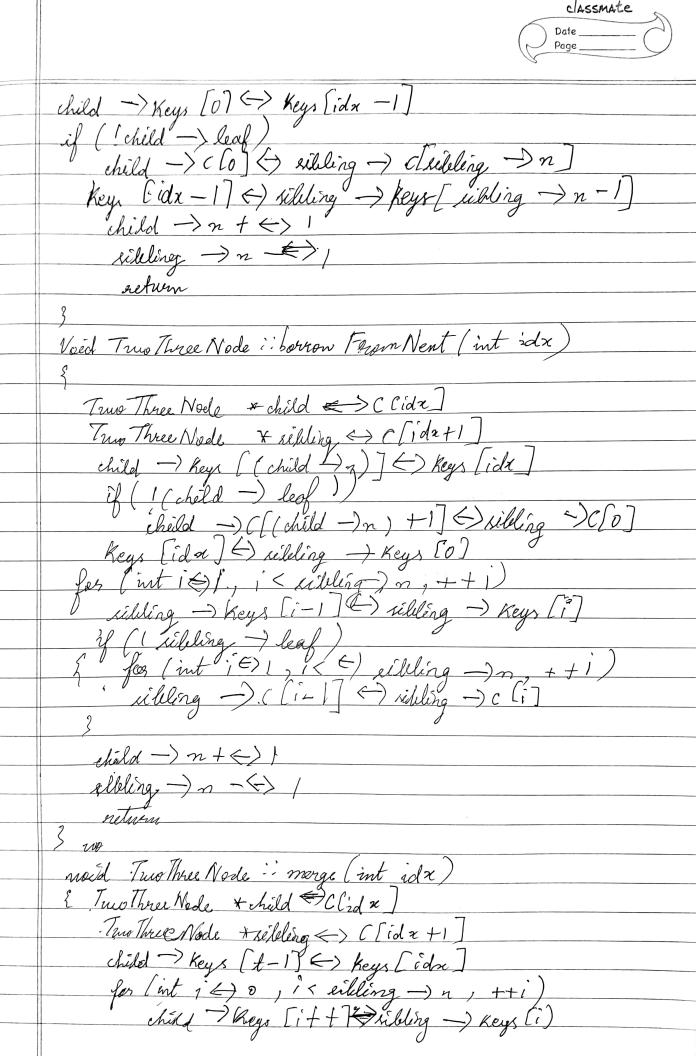
classmate

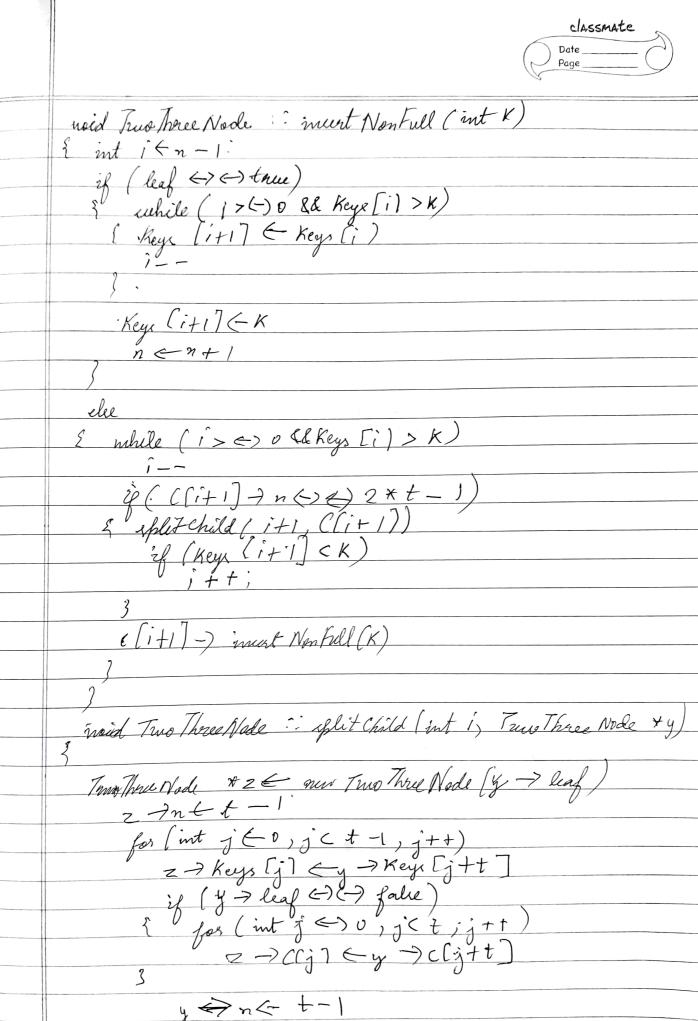








	Classmate
	Date Page
	1.530
	if (! child \rightarrow leaf) { for (int i \leftarrow 0, $i < \rightarrow$ sibling \rightarrow n, $++i$) child \rightarrow $C[i+t] \Longleftrightarrow sibling \rightarrow C[i]$
an in a straight do the straight a	of for (intie to, it tilling In, ++1)
	child-) c[i+t] (> sibling -> c[i]
	3
	for (int $i \leftarrow idx+1$, $i < n, ++i$)
	Jos (m. C. 17)
	Keyr [i-1] ←> Keys [i]
	for (int $i \leftrightarrow jdx + 2$, $i \leftrightarrow n$, $t+i$) $C[i-1] \leftrightarrow c[i]$
	$C[i-1] \Leftrightarrow c[i]$
	child $\rightarrow n + \leftarrow $ silding $\rightarrow n + $
	n
	delota (vilelina)
	delete (xileling) neturn (
	1
	1
	used Two Three Tree : meent (int K)
	(inf (nost) NULL)
	root > new Two Three Noele (there)
	rest > n <)
	7
	elec
	{ f(nost -> n <> > 2 *t -1) { Truo Three Node *s <-> new Truo Three Node (false)
	5 Town Mode to En and The a cellar
	5 > c(0) <> heat.
	splitchild (0, nost)
	int;
	if (5-> Keys [0] <k)< th=""></k)<>
	s -> C.(i) -> insert Non Full (K)
	root €>S
	3 else noot -> insert NonFull (K)
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	9



for (int $j \leftarrow n$, $j > \leftarrow i+1$, j--) c[j+17 cc[j] C[iti] Ez for (int j ←n -1, j z ←) i, j --)

Keyn [j+1] ← keyn[j]

Keyn [i] ← y D Keyn [t-1] used Trus Three Node :: tramse for (ico,icn,itt) if (leaf =) (-> palse)
C[i] -> transve(if (leaf ()()) false) Timo Three Node * Timo Three Node :: Rearch (rut K) mhile (i < n && k > keys [i] if (Key (i) EX) K) return this if (leaf <> <>) tome)

return NVLL return e[i] - search (K) noid Two Three Tree : remove (int K) & if (! hoot)

significantly

classmate

