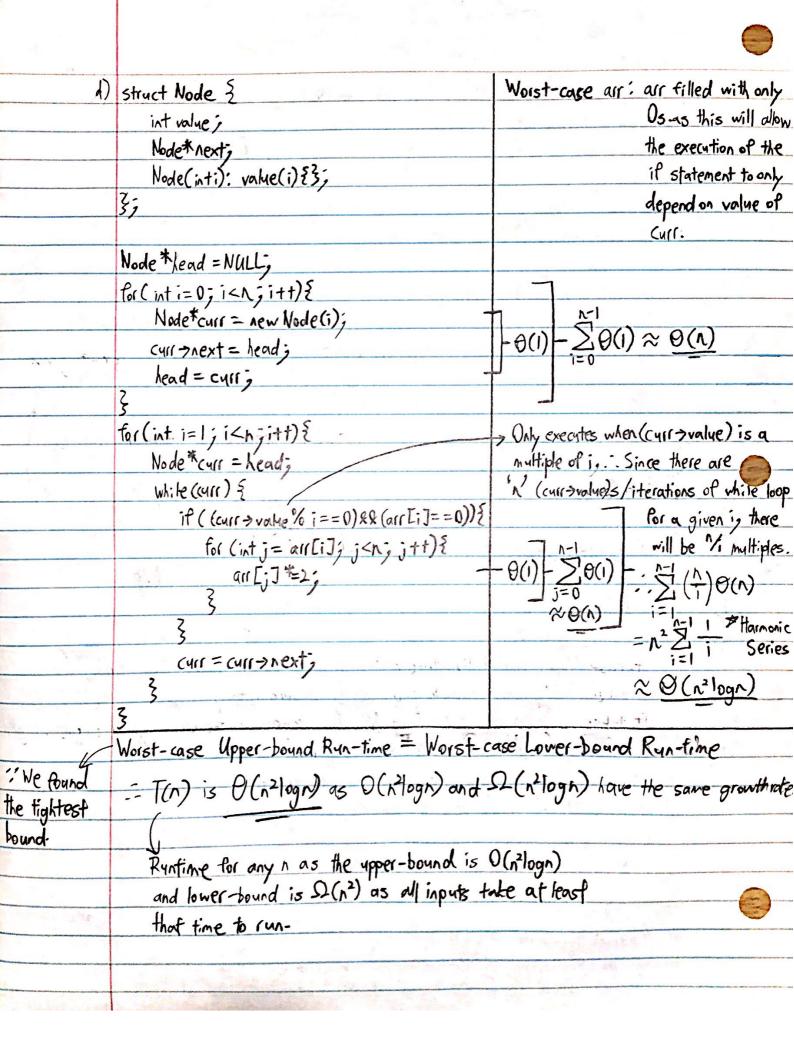
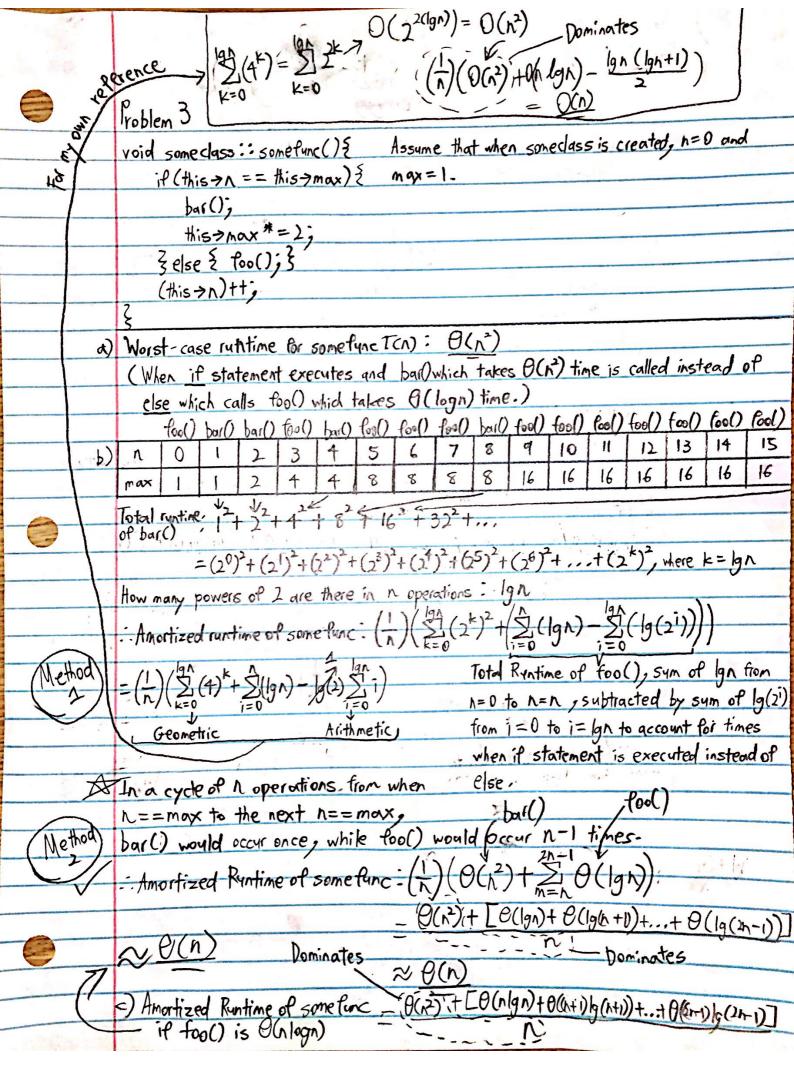


Queue q; //all operations on a Queue take constant time. for (int i=1; i<=n; i++)} $-\sum_{i=1}^{n} \Theta(i) = \Theta(n)$ q-enqueue(i); 0(1) bool swap = false; while (!q. empty ()) } if (swap) } if (q, front () = = 1) { for (int i= n+1; i<= 2n; i+t) { ZO(1)≈O(1) q. enqueue (i); q-dequeue(); U(1) . Removes one integer from que ae For every m # of integers in the queue, it will take in times to remove if from the queue as one integer is else { q. enqueue (q. front()); O(1): Size of queue remains the same g-dequeve(); Swap = !swap; Worst-case Upper-bound Run-time = Worst-case Lower-bound Run-time 1 It takes O(n) at the start to enqueue n integers into queue-: We found the 2. It takes $\theta(2n)$ to degrepthe first 1 to n integers in the guereas a single integer tightest bound. is dequeued hat the time the while loop ryns as explained above. 3. If takes O(n) to exqueue the next h+1 to 2n integers in the queue after q. floot()==1 4. It takes O(2n) to dequeue the htl to 2n integers in the queue as a single integer is dequeved half the time the while loop runs as explained above. 5- while loop ends when grempty() == true. : T(n) is $\theta(n) + \theta(2n) + \theta(n) + \theta(2n) \approx \theta(n)$ as $\theta(n)$ and $\Omega(n)$ have the same growth rate.) Runtime for any n as the upper-bound is O(n) and lower-bound is $\Omega(n)$ as all inputs take at least that time to run-





Worst-case sequence. d) void someclass: another func() { n=0, max=1if (this > n > 0) { somefunc () -> foo (); (this > 1) --; n=1, max=1somefunc() > bar(); if (this >n < (this >max)/2){ n=2, max = 2bar(); - 0 (n2) some func() > bar(); { this > max /= 2; 1=3, max =4 3 else { foo(); 3 - O(logn) another func() > foo(); Gole h=2 , max =4 another func() -> bar(); the worst-case sequence happens when somefund, followed by anotherfunc(), followed by anotherfunc is called when h==maxas bar is executed twice in each cycle and once for fool, -: Amortized runtime / function coll: O(n2) + O(lg(n+1))+ O(n2) $\approx \theta(\kappa^2)$