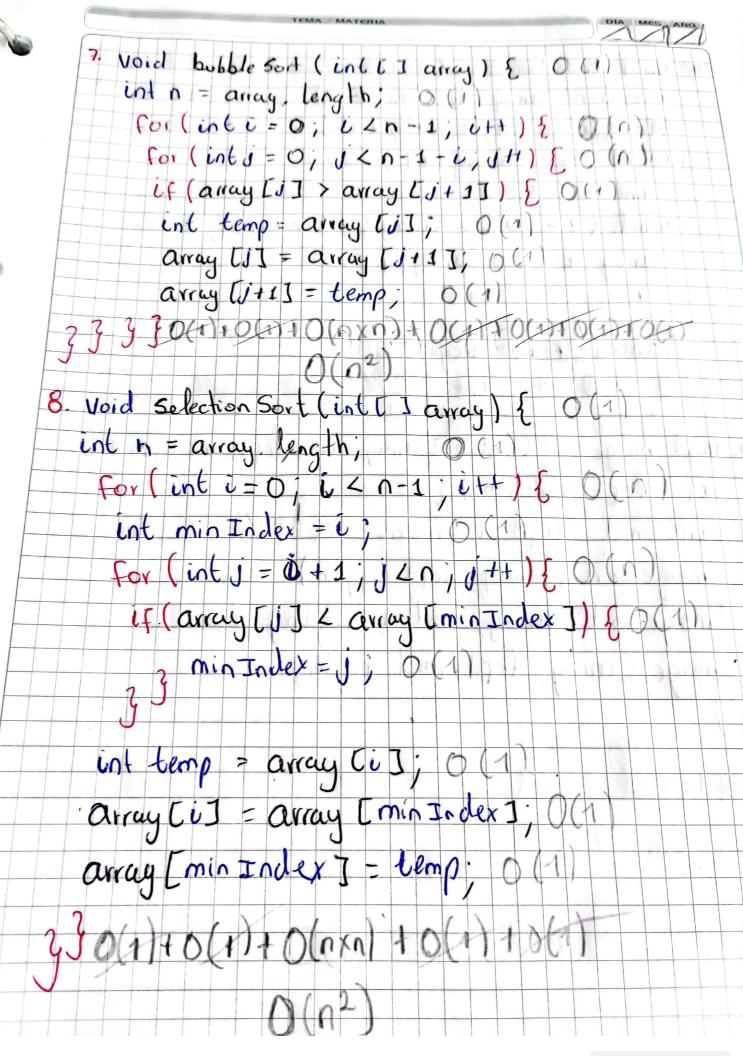
506222015 Brain Felipe Range Oilis Analisis Big C 1 for (int i = 0; i < n; i ++) O(n) o(r)2. For (int i = 0; iz n; itt] [(n) for (int j=0, JZm, j++) { (m) 0(n) x 0 (m) $O(n \times m)$ 3. for (int i=0; i < n; i+1) { O(n) for (int j = 0, JKn; jtt) { $O(n) \times O(n) = O(n^2)$ 4 int index = -1; for (int i= 0; i2 h; itt) { 0 (h) if (array[i] = = turget) index = i; 0(1 Break; 0(1100)10(1100) 0(n

int legt = 0, right = n-1, index - -1; o(n) int mid = leg + + (nyth - lest) /2; O(lyn) if (anay [mid] 4 = target) { index = -anid; OCO Break; I de it (array [mid] 2 target) [00 legt = mid + 1; Jelse { eigth = mid - 1; 01 30(n) 0(m) 0(m) 1000 1000 1000 1000 O(togn) 6. Int row =0, col = matrix [O]. Length -1, index Row -1 index Col: -1; 0(1) while (row < matrix. length && cal >= 0) { 0 (1) if (matrix [row][col] == target) { O (n) m index Row = Row; O(n) index Col = col; O Break; O (1 3 else if (matrix [row][col] & target) { O(n+m) Col - - 10 0(1) + 0(m+n) + 0(n) + 0(n) + 0(1) + 0(n+n) + 0(n) O(mtn)



9 Void insertion Soil (int (I array) { int n = away length; O(1) (or (int i = 1; i < n; itt) { O(0) int key = array [0]; O(1) int j = i - 1; 0(1) while (1) > = 0 & & array (1)] > Key) { 0(1) array (i 1] = array []; O(1) array [j+2] = Key; O(1) 3 30(1) + O(n) + O(n) + O(n) + O(n) + O(n) + O(n) 10. Void merge soit (int () array, int left, intright) { O(1) if (left (Right) { O(1) int mid = left + (right - left) /2; Ollogn merge Sort (array, left, right); O(1) maye soit (array, mid 11, rigth); merge (array, left, mid, right); O(1 0(1)10(1)10((690)10(1)10(1)

