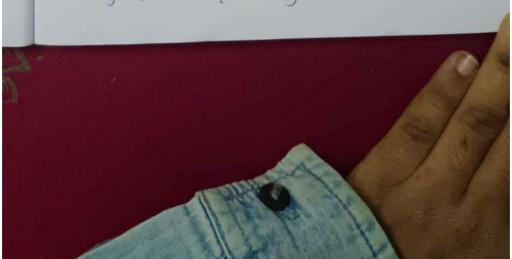
lase Sheety 1) Write an algorithm that suit to solve the problem? Thep 1: - Divide the pile of socks into two roughly equal parts Step 2 : - Each group of campen independently sort their . respective piles of socks wing any officient sorting olgorithm step3: - merge the 2 sorted piles into single sorted pile, Combine the sorted piles in a way that maintains the sorted order, Use a merge algorithm, similar to the one used in Mergesort stept :- 4 there are more than I pile, continue step 1,2,3 agan. step 5 : - The find result is a completely sorted list of saks 2) Determine the no: of phases needed for PIP to arrange the socks for faster wash The noist phases needed for PLP is determined by the nosof recursive calls in the algorithm. Each recursive call represents a phase to this case, since the algorithm divides he piles into 2 halves at each level, the mosof phase is log_(n), where n is the nosof socks. 3) Write a pseudocode algorithm outliving the sorting process, considering individual corting & merging phase. function PLP_Sort (socks): if length (socks) ==1: veturn socks mid = leng th (socks) /2 left-pile - socks [o to mid-1] right-pile = socks (mid to end)



lest-pile = PIP_Sort (left-pile) right pile - PIP-Sort (right pile) result = merge (10st-pile , right-pile) vehin result function Merge (left, right): result = [] left_index=0 right_"index=0 while left_inder < length (left) & right_index < length (right). if left (left - index) + right [right - index]: result. append (left [left_index]) 10St-index+ else: result, append (right (right index)) right_index++ result . extend (left [left - index = 7) result . extend (right [right_index = 1) yohun result 4) Derive the recurrence relation of above algorithm? The recurrence relation for the given algorithm is: $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + O(n)$ recurrence relation describes the time complexity of the PLP sorting algorithm. The 2.T(2) term represent the time to sort & merge 2 sub-piles, 2 0(n) represents the

time to merge the 2 sorted sub-piles