

Algorithm

Step 1: Start

Step 2: Input $a[] = \{64, 82, 95, 105, 15, 26, 8, 1, 2, 25, 2, 5\}$

Step 3: merge (int a , int i_{begin} , int i_{mid} , int i_{end} , int b)

Step 4: splitting merge (int b , int i_{begin} , int i_{end} , int a)

Step 5: merge sort (int a , int $size$)

Step 6: int $size = size\ of\ a / size\ of\ a[0]$

Step 7: int $ret = merge_sort(a, size)$

Step 8: Display sorted array elements

Step 9: for (int $i = 0$; $i < size$; $i++$)

Display $ret[i]$

Step 10: free (ret)

Step 11: Stop

merge (int a , int i_{begin} , int i_{mid} , int i_{end} , int b)

Step 1: Entry

Step 2: int $i = i_{begin}$, $j = i_{mid}$

Step 3: for (int $k = i_{begin}$; $k < i_{end}$; $k++$)
 $b[k] = a[i_{mid} & k] \ (j = i_{end} \ \& \ a[i]k = a[j])$; $i++$; $j++$

Step 4: End

Splitmerge (int ~~a~~ b, int i begin, int i end, int ~~a~~ a)

~~Step 1~~ Step 1: Entry

Step 2: if (i end - i begin < 2)
return

Step 3: int mid = (i end + i begin) / 2

Step 4: Splitmerge (a, i begin, i mid, b)

Step 5: Splitmerge (a, i mid, i end, b)

Step 6: merge (b, i begin, i mid, i end, a)

Step 7: End

merge sort (int ~~a~~ a, int size)

Step 1: Entry

Step 2: int ~~ret~~ ret = malloc (size * size of ~~ret~~ ret)

Step 3: int ~~temp~~ temp = malloc (size * size of ~~temp~~ temp)

Step 4: for (int i = 0; i < size; i++)

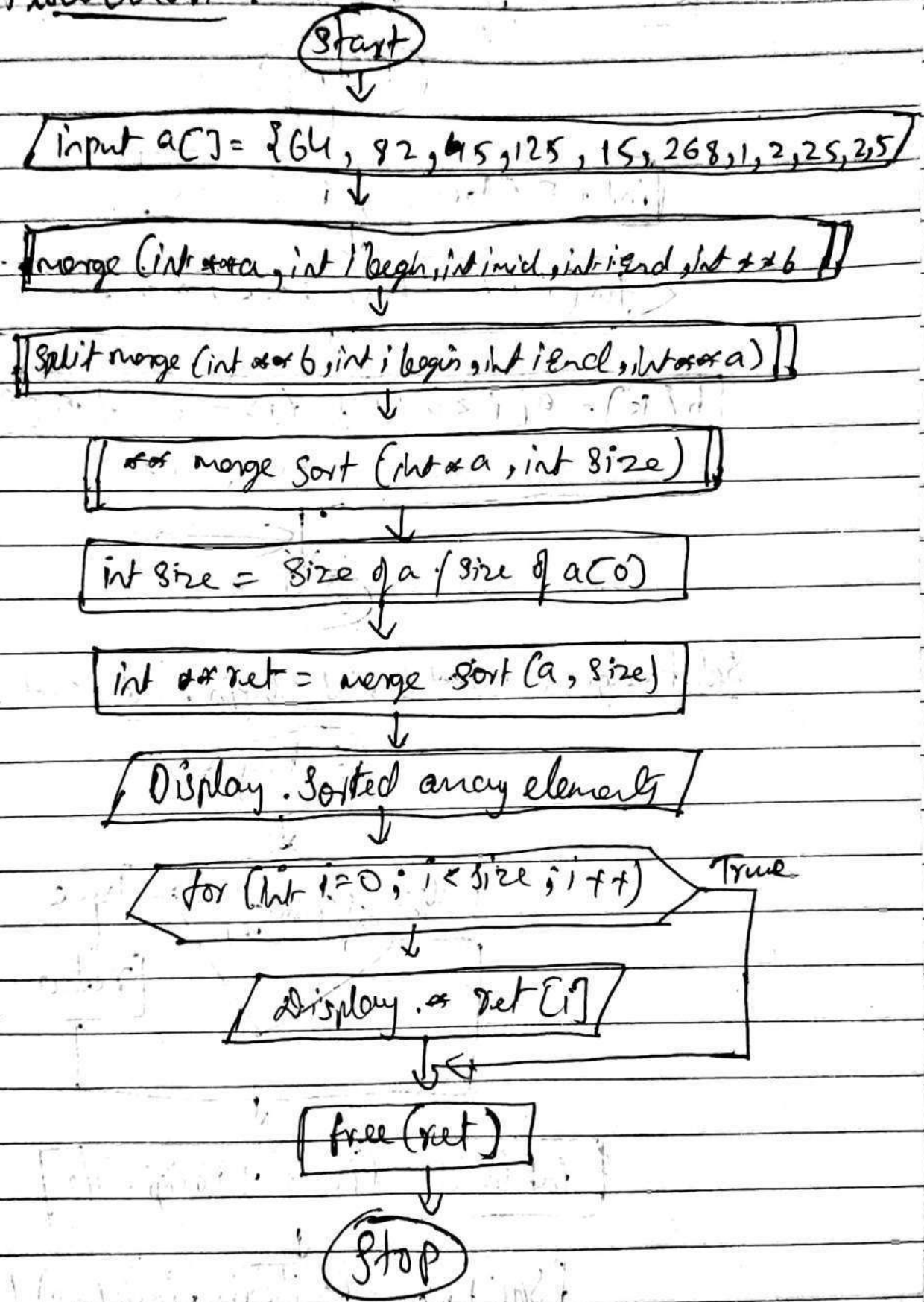
ret[i] = temp[i] = a + i

Step 5: Splitmerge (temp, 0, size, ret)

Step 6: free (temp)

Step 7: return ret

Flowchart



Merge (int ~~a~~, int i begin, int i mid, int i end, int ~~b~~)

Entry

int i = i begin; i = i mid

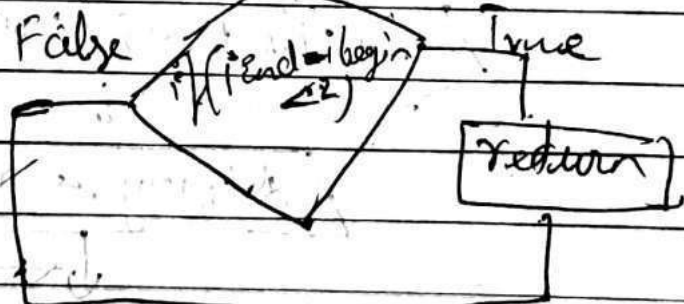
for (int k = i begin; k < i end; k++)

b[k] = a[i < i mid && (i >= i end || a[i] <= a[i]) ;
i++; i++ ..

End

Split merge (int ~~a~~ b, int i begin, int i end, int ~~a~~ a)

Entry



int i mid = (i end + i begin) / 2

Split merge (a, i begin, i mid, b)

↓
~~merge~~ (b, i, begin, i)

↓
Split merge (a, i, mid, i, end, b)

↓
merge (b, i, begin, i, mid, i, end, a)

↓
End

merge sort (int *a, int size)

Entry

↓
int *ret = malloc (size * size of *ret)

↓
int *temp = malloc (size * size of *temp)

↓
for (int i = 0; i < size; i++)

↓
ret[i] = temp[i] = a + i

↓
Split merge (temp, 0, size, ret)

↓
free (temp)

↓
return ret