$$n^{19} x^{3} = n^{19} x^{16} = n^{2}$$
  $n! > n^{2} \implies f(x) = \sum_{i=1}^{n} (n^{10} x^{16} + E)$ 

$$\int_{1}^{1} 3 x^{2} = \int_{1}^{1} \frac{1}{3} \int_{$$

c) 
$$871\%$$
) +  $4n^3 = 7(n)$ 

$$f(n) = O(n^2)$$

can't be solved with moster theorem because f(n) must be

1-) f) T(n) = 2 T(n/2) - n can't be solved with matter therem because f(n) = -n must be positive.

gl T(n)= 3 T (n/3) +10/1 can't be solved with matter theorem because 1010 is not Polynomial.

2) a)  $T(n) = 9T(n/3) + n^2$ a=5  $f(n)=n^2$   $n^{(0)} = n^2$   $f(n)=O(n^2)=O(n^{(0)} + 3^5)$ 

So  $O(n^{10/6^2} \cdot h_{0/n}) = O(n^2 \cdot l_{0/n})$ b)  $T(n) = 8 T(n/2) + f(n) f(n) = O(n^3)$ 6=2  $n^{1913}=n^3$ 

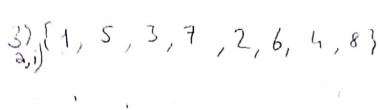
So O( n'ose . loga) = O(n'.loga)

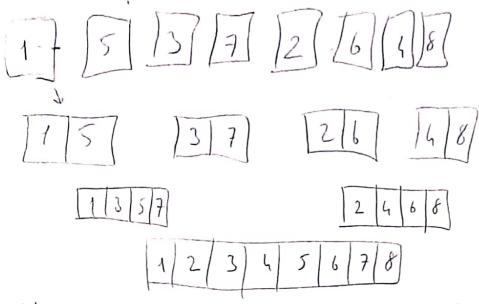
c) T(1) = 2 T(1/4) + f(1) N/012 = N/032 = N/2 = JA

fu) = O(50)

So O (nor. logn) = O (In. logn)

I would choose c. Because it is the fastest among them. And in today's circumstances, memory is not a problem.





This way every number will be compared with each other. And this is the way of maximum number of comparisons.

a.ii) { 1, 2, 3, 4, 5, 6, 7, 8}
Already sorted arrays will require minimum number of comparisons

Falia DaFa4

h) The algorithm salves problems by dealing with only one half of the array.

Because of if-else black, subproblem number is one. Because only if or else is working.

f(n) = O(1) because assigning is O(1) b=2 - half the size

T(n) = 1. T(n/2) + 1

 $n^{1}95^{2} = n^{1}912^{1} = n^{0} = 1$   $f(n) = \Theta(1)$ 

So O(nogo 10gn) = O(no. 1yn) = O(1.1yn) = O(1.1yn) =