

a)so if you look the value of i is squared every time so it starts with 2,4,16,256....

So this complexity we can take it as $\log \log n$ so the complexity is $O(\log \log n)$

b)here the outer loop runs n times and here in the inner to compute square root it takes $\log n$ times and the inner loop runs at most n^3 times so the final complexity is $O(n * \log n * n^3) = O(n^4 \log n)$.

c)so the top loop runs n times and then loop below it also runs times and now the loop inside is doubling every time is it goes like 1,2,4,8.. n so which is $\log n$ in total so the total complexity is $O(n * n * \log n) = O(n^2 \log n)$

d) he runtime is $O(n)$ since you are looping through the input array once. considering the cost of each operation in the loop (i.e. the cost of creating a new array, copying over the contents of the old array to the new array, and deleting the old array) , we get that the the total cost of all these operations is $O(n)$. Thus, the overall runtime of the function $f()$ is $O(n)$.