

2-)

Se asume que se usa median para conseguir el index de la mediana, y luego se utiliza una función similar a partition, llamada median modificado, el cual separa el array en dos sublistas a través del uso de la mediana

a-) El mejor caso sería que se encuentre la mediana en la primera iteración, lo cual de todas formas tendría una complejidad de  $O(m)$ , pues debe hacer un partition, lo cual revisa todos los elementos una vez.

Dado que solo se tiene el index de la mediana, se debe aplicar un nuevo partition usando la mediana, lo que nuevamente costaría  $O(m)$ .

Finalmente, quicksort repetirá este proceso una cantidad  $O(\log m)$  veces.

Entonces en el mejor caso, esto costaría

$$O(2m \log(m)) \approx O(m \log(m))$$

b-) El peor caso sería cuando al buscar la mediana, se tome el máximo número de iteraciones, lo que costaría  $O(m^2)$ . La partición seguiría siendo  $O(m)$ .

Quicksort también repetirá esto  $O(\log m)$ .

Entonces el peor caso tendrá complejidad

$$O(m^2 \log m + m \log m) \approx O(m^2 \log m)$$

c) Yo le diría que no es una mejora razonable, ya que

el mejor caso iguala al caso promedio de quick sort, y el peor caso es peor que el de quick sort, por lo tanto, o se demone lo mismo o se demone más, por lo que no vale la pena usarlo.