Q1:

if we consider 1 to be an exact power of 2 then from [1..n] we have $\log n + 1$ exact powers of 2.

So,

$$T(n) = (n - \lg n - 1) + \sum_{i=0}^{\log n} 2^i$$

or $(n - \lg n - 1) \leq n$

and $\sum_{i=0}^{\log n} 2^i = 2^{\lfloor \log n \rfloor + 1} - 1 \leq 2^{\log n + 1} = 2n.$

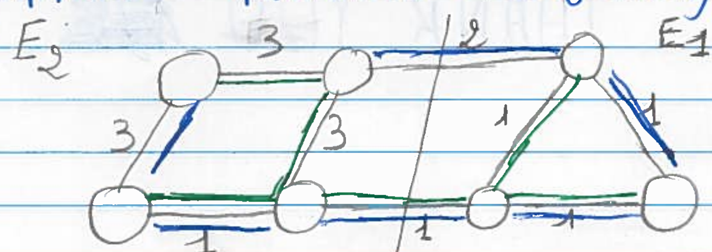
$\Rightarrow T(n) \leq 2n + n = 3n$

$T(n) = O(n)$

which gives $O(1)$ amortized cost by operation.

Q2:

The algorithm proposed by prof Toole is not correct. Here is a counter example. let's have the following graph and we partition it in the following manner.



The result of MSN by prof Toole alg is shown in green

~~but if we use Kruskal's alg~~

The sum of the egges weight of the edges is $1+3+3+1+1+1+1 = 11$.

but if we use Kruskal's algo, the result is shown in blue the sum of the weight of edges is $1+1+1+1+2+3 = 10$

$\Longrightarrow$ The algorithm proposed doesn't correctly compute THe MSN of a graph.

Q3:

worst case:

Randomized-Quicksort will return the index 0 or (n-1) with n being the length of the array. at each time, one recursion call is made resulting in $\Theta(n)$ calls to random number number generator.

Best case:

Randomized-Quicksort will return the pivot resulting in dividing the array into 2 equals-lenght sub arrays. The recursion tree will be log(n) levels deep. -> double the number of recursions. -> giving $2^{\log n} - 1$ recursions.

$\Rightarrow 2^{\log n} = n.$ $\Rightarrow \Theta(n)$ calls to the random number generator.

Q4:

THANK YOU