2) Em N operações de custo $c(i) = \begin{cases} i & \text{se } i \text{ é potência de } 2 \\ 1 & \text{c.c} \end{cases}$

o custo amortizado é:

$$\sum_{i=1}^{N} c(i) = \sum_{\gamma=0}^{\lfloor \lg N \rfloor} 2^{\gamma} + \sum_{\lg(i) \notin \mathbb{N}} 1 \leq \sum_{\gamma=0}^{\lfloor \lg N \rfloor} 2^{\gamma} + \underbrace{N}_{\text{limitante superior}} = 2^{\lfloor \lg N \rfloor + 1} - 1 + N \leq 2^{\lg 2N} - 1 + N$$

$$= 2N - 1 + N = 3N - 1 \leq 3N$$

Assim, o tempo amortizado por operação é constante.

9)

Pela heurística dos ranks, temos que a altura da árvore é limitada por uma constante c*lg n.

Pela compressão de caminhos e a heurística dos ranks, temos que as operações UNION(LINK) e FINDSET têm custo amortizado O(lg*n), tal que lg*n = min{ i tq lg^(i) n < 1}, portanto é assumido como constante por operação.

Então, dada a sequência de M operações MAKESET = O(1), LINK ~ O(1) e FINDSET ~ O(1), temos que no pior caso tempo consumido é O(M) pois cada operação é de ordem constante. (Note que o LINK ser sempre realizado antes do FINDSET é o que possibilita esse comportamento porque ele atualiza o campo parent e rank de cada nó).

Se apenas a compressão de caminhos estiver implementada, a operação FINDSET consome O(N) tal que N é o número de elementos por que não há como garantir o balanceamento da árvore e a operação UNION(LINK) faz duas chamadas de FINDSET. Então, uma sequencia de M operações pode consumir O(M + N) porque M representa a soma total de todas as operações feitas entre MAKESET, LINK e FINDSET, de tal modo que, embora a última não seja da ordem de log(n) ainda há a amortização garantida pela compressão de caminhos.