1.(a) I would divide n dwarves into pairs (if n is an odd number and one dwarf is left, then I would pair up with him), and then one member in each pair would help the other member to enter a barrel, which means half of the dwarves would enter barrels during this process. Afterwards, I would pair up the remaining dwarves and repeat this process, until there is only one dwarf left. Then I would help him enter a barrel first, and finally enter my own barrel.

Since we pair up dwarves in each iteration, the total times of iteration is $\lceil \log_2 n \rceil$. Since each pair could enter barrels simultaneously, total time in iterations is $10 \cdot \lfloor \log_2 n \rfloor$. After iterations, we need extra 20 seconds to let the remaining dwarf and myself enter barrels, so the total time to escape is :

$$T(n) = 10 \cdot \lceil \log_2 n \rceil + 20 \in O(\log n)$$

b (i) Since we do this in serial, one element would be added 1 after another. Assume add 1 takes a constant time c, total time :

$$T(n) = c \cdot n \in O(n)$$

(ii) Since we do this in parallel, every element could be added 1 at the same time, Assume add 1 takes a constant time c, total time :

$$T(n) = c \in O(1)$$

(iii) Since we do this in serial, we should add every element one after another, which means we can only add one element at the same time. Assume add one element takes a constant time d, total time:

$$T(n) = d \cdot n \in O(n)$$

(iv) I would divide n elements into two groups in the middle, and then divide each group into two groups in the middle again, repeat this process until each group only contain one or two elements, and then add them together from small to big groups:

First, set left boundary, left, to be 1
set right boundary, right, to be n
then start SumUp

```
SumUp (left, right)
    if left = right
        return left
    else if right - left = 1
        return left + right
    else
        return SumUp (left, (left+right)/2) + SumUp ((left+right)/2 +1, right)
```

(v) Since we break elements into two ~~pairs~~ groups in each recursion, the total times of recursion is $\lceil \log_2 n \rceil$. Assume add one element takes a constant time d, and we do this in parallel so that at each recursion we can add elements at the same time, so the total time:

$$T(n) = d \cdot \lceil \log_2 n \rceil \in O(\log n)$$