Question 13.

(i) A disjoint set has at least 1 item. If its parent pointer has been updated once, it has been merged with another set, so the new set size is at least 2. On another merge, due to the weighted union heuristic, if the item's parent pointer has been updated it belongs to the smaller of the two sets, so the other set must have had at least 2 items too, so the new set has at least 4 items. This logic repeats, so after k parent pointer updates, the set must have at least 2^k items.

(ii) Consider two items 1 and 2. When they are merged, the parent pointer of 2 updates to 1. Then consider 3 and 4. 4's parent pointer updates to 3 on merge, and merging these two new sets, the parent pointer of 1 could update to 3. Then, the parent pointer of 1 would not update until another 3 merges have occurred to create another set of size 4, which this set could merge into, becoming a set of 8. Then, another 7 merges have to happen to create a new set before 1's parent pointer can update again, and so on. 1 has the most updated parent pointer and it is bounded by log_2(n), so the number of times each item has its parent pointer updated is also bounded by log_2(n), i.e. O(logn).

(iii) Let Φ = no. of items in max size tree.

<u>get_set_with</u>

Just looks up parent pointer.

$C = O(1)$

$\Delta\Phi = 0$ as no change to items.

$\therefore C + \Delta\Phi = O(1)$

<u>add_singleton</u>

Just puts new item, no changes to pointers.

$C = O(1)$

$\Delta\Phi = \begin{cases} 1 & \text{if no other elements} \\ 0 & \text{otherwise} \end{cases}$

$C + \Delta\Phi = O(1)$

<u>merge</u>

$C = O(N)$

$\Delta\Phi = O(N)$

$C + \Delta\Phi = O(N)$

NOT SURE WHERE TO GO FROM HERE