

## COMP3121 21T2 Assignment 2 Q3

Written by Zheng Luo (z5206267)

In order to kill all monster, the key is survived by killing monster who has highest unit of strength as feedback. Hence let the different between strength cost and strength feedback of killing a monster as  $b_i$ .

$$b_i = a_i - g_i \quad (1)$$

Sort  $b_i$  in decreasing order by using merge sort with time complexity of  $O(i \log(i))$ . Create a linked list *diff* with size of  $i$ , each node consists the information of  $b_i$  and  $a_i$ , the linked list will connected in descending order based on the value of  $b_i$ , hence the monster with most rewarding strength will be located in the front of list. The time complexity of inserting and removing are  $O(1)$  for each element.

Examine the list in order, kill the  $i_{th}$  monster if  $currentNode.a_i < currentHealthS$ , otherwise move to next node. Once the monster has been killed, update the current health  $S$  and remove current monster node, and re-examine the list from head, repeat the process until list is empty. Return the order of monsters in the original list if the health  $S$  is greater than 0 when list is empty, otherwise return "no such ordering".

The overall time complexity is  $O(i \log(i))$  since  $O(i \log(i))$  dominates  $O(i)$  for inserting and removing the element in list.