

# COMP3121 Assignment 1

## Q2

### 2.1

To select a banker to remove while maximising profitability, DAC investments will have to remove the banker with the lowest contribution to the company's overall profitability.

The CEO's nephew will be represented in the given integer arrays  $M[1..(k + 1)]$  and  $P[1..(k + 1)]$ , where the last entry,  $(k + 1)$ , is the CEO's nephew.

Thus, DAC should remove the banker from 1 to  $k$  managing the lowest dollars' worth of investments. Only selecting the banker from 1 to  $k$  will ensure that the CEO's nephew is not removed. The CEO's nephew has the lowest performance rating, so any banker removed won't affect the  $\min(P)$  section of the formula and won't affect the overall profitability of DAC Investments.

By removing the banker managing the lowest dollars' worth of investments, it is ensured that the first part of the equation (total sum of investments) can be the highest value.

This will result in the maximum profitability as the profitability equation relies on the total sum of investments multiplied by the lowest performance rating of the workers which will always be the nephew.

### 2.2 SKIPPED

### 2.3

Consider the algorithm,  $f$ , which will select  $k$  members to be taken from  $n$  employees that maximises the profitability of DAC Investments. This algorithm must also run in  $O(n \log(k))$  time.

The overall profitability of DAC investments is given by

$$(M[1] + \dots + M[k]) \times \min(P[1], \dots, P[k]).$$

The time complexity of this equation is  $O(n)$ .

The algorithm should iterate through both  $M$  and  $P$  with the index  $i$  where the values of  $i$  are bounded by  $[1, k]$ .

During each iteration of both arrays, find the product of each banker's investment value  $M[i]$  and performance rating  $P[i]$ . The values of these products should be store in a new array of objects,  $A$ , which will store the banker's profitability contribution (*profitability*) as well as their index (index) in the  $M$  and  $P$  arrays.

This will give a surface level value of the banker's contribution to the profitability of DAC.

To choose which  $k$  members from  $n$  bankers, sort the array,  $A$ , in descending order of  $A.profitability$ .

Then, iterate through  $A$  and select  $k$  number of bankers.

This will result in a selection of the  $k$  number of bankers with the highest overall contribution to profitability.

The time complexity of this algorithm will be  $O(n)$  as the worst case time complexity of iterating through an array is  $O(n)$  and the operations of multiplication are  $O(1)$  time.

Thus, the time complexity of this algorithm to select  $k$  members out of  $n$  bankers is  $O(n)$ .