**Not The Corona Virus Team Member’s Information**

|  |  |  |
| --- | --- | --- |
| **Team Member Name** | **Email Address** | **Challenge** |
| Cesar Diaz | Calexdiaz143@gmail.com | 1 |
| Rhodora Villanueva | Rhodora33.rv@gmail.com | 1 |
| Jaimie Gillette | Jaimie.gillette@gmail.com | 2 |
| Timmy Ngo | Timmyngo021@gmail.com | 2 |
| Randy Rodriguez | Randy.renee22@gmail.com | 3 |
| Carlos Sanchez | Sanchezcj95@gmail.com | 3 |
| Annie Voigt | a.voigt1121@gmail.com | 3 |

**Overview of Approach**

For Challenge 3, we take the sailors who will die without treatment (that is, Health + rate of health decline, or T, < 0), and give the 30 healthiest of these a bed. Those with the lowest T values who have a bed receive the reusable resources, and then non-reusable resources are allocated to those who are still dying based on T values.

Those who have the highest negative T values receive non-reusable treatment first. The last step is dependent on the day, as on day 3 we want to use all our leftover resources. The first two days, we only give non-reusable resources until all sailors have positive T values. The third day, after all sailors have T values above zero, we allocate the rest of the non-reusable resources based on whose T value is lowest.

**Technical Description of Solution**

We use two sorts on the dataframe in order to determine who receives treatment first. The first sort is whether or not T < 0, and the second sort is on the absolute value of T. This ensures that we helped the sailors most likely to survive first with T < 0, and then after all sailors with T < 0 are given resources, we give resources to those such that T > 0, sorting T lowest to highest

It is important to note that after each treatment is given, we re-sort the dataframe to find the new health values.

For the non-reusable resources, we conserve as many resources as we can so that we save the maximum amount of sailors. We use a combinations function to create all given combinations of medications available and from that list choose the minimum possible health boost that would get T > 0.

After every sailor with a bed has T > 0, or all the resources are gone, we return the results. On the last day, we use an extra step to allocate the rest of the resources, by determining who has the lowest T score and giving them the treatment with the greatest health bonus, and continuing down the list.

We have an alternative solution that takes the approach of only allocating beds if it is determined that the sailor can be saved with the available resources. This is done by first checking whether the amount of sailors where T < 0 was greater than the amount of beds available, as we did before. Rather than assigning the beds at that point, however, we first attempt to find a combination of resources that could get the final health of the sailor above 0. If no combination is found, we deem the sailor as unsavable and do not give them a bed. We then go through every dying sailor and go through all possible combinations of resources, both reusable and non reusable, that will just get the sailor’s health above 0. We also make sure to prioritize the use of reusable resources if they are available when considering combinations, as well as how much of the available resources there are. After making sure that no sailors are dying, we check to see if we have leftover beds and reusable resources and give these to the sailors with the lowest health. The allocation of resources on the final day was unchanged from our first solution. We did not use this alternative solution’s results for our final allocations because the score was slightly lower.

**Reason(s) for Choosing the Approach Taken**

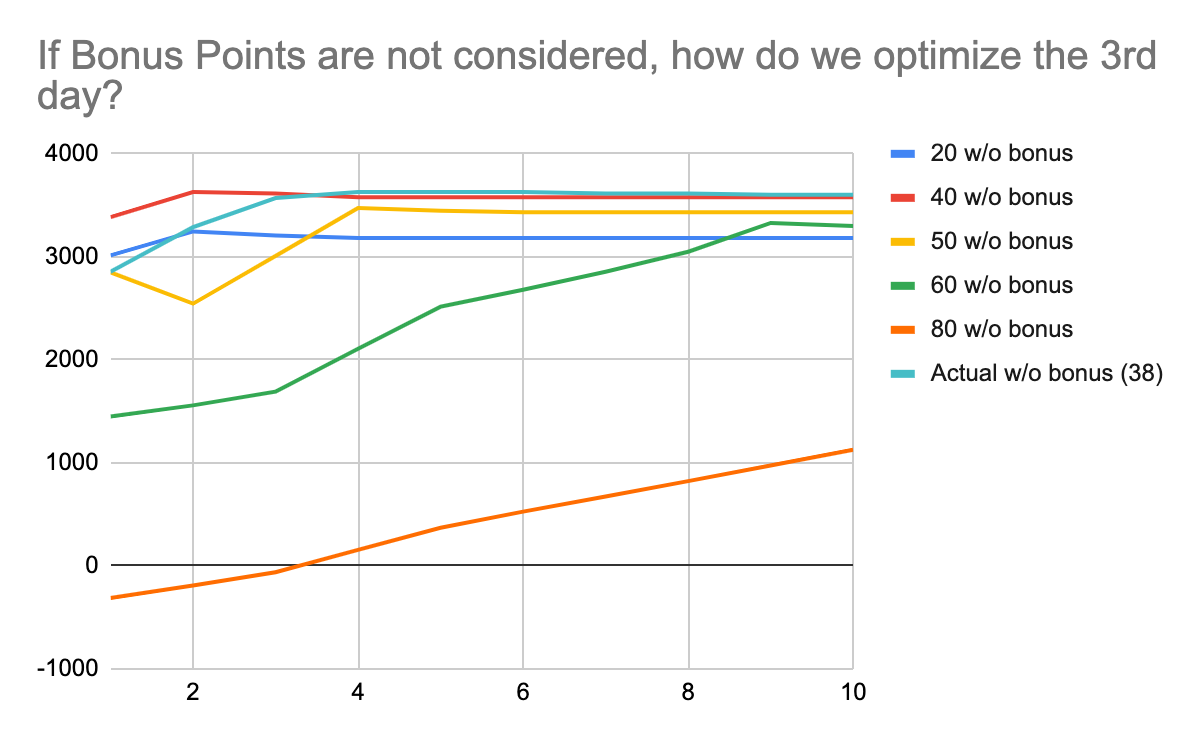
We choose this approach because we understand that this problem does not require a more complicated approach. We choose to allocate resources to healthiest sailors with T < 0 because we want to save as many sailors as possible. Beds are only allocated to the healthiest 30 individuals because that will give us the lowest amount of health needed to reach a value above 0 for each of the sailors we treat. We use the combinations function in order to consider every possibility of resource allocation possible to get the sailors T value above zero, without wasting resources that can be used to save other sailors.

Throughout our brainstorming process, we considered the possibility that some sailors would not be saveable if we had more limited resources. In this scenario, we considered building a function to take those sailors out of the beds and give their spot to sailors with T > 0 in order to boost our final points. However, there were more than enough resources so we did not have to create this function.

We found it difficult to predict how many sailors would be given to us in future days. Without this knowledge, it was hard to predict how many resources we should be stockpiling. On the second day, we created a function to determine our final points based on how many resources we decided to save and how many sailors we were given on the third day. We realized this would not be relevant because we would receive half-points each day for reserving the resource, therefore, if we had too many resources on the third day, saving them would result in us receiving the same amount of points as if we had used them efficiently.

So we ask the question: if we did not receive bonus points, how would we optimize how many resources to save for the third day?

The following graph takes into account what our final points would be if we saved n number of resources per resource type, and we didn’t take into account bonus points for saving the resources. Each line symbolizes a different number of sailors given on the third day, as we were not given this information. The datasets were randomized for the sailors, giving them a health number in (1, 100), and a health decline number in (0,-99).



Some lines, such as 20 and 40, peaked very early which makes sense as we would just be wasting resources if they had been saved to the end. However, these peaks are very minimal and so we should always be reserving as much resources as we can.

For allocation of resources, we envisioned a scenario where a sailor could not be saved, but had we administered a different resource to a different sailor, that sailor could have been saved. We considered creating a recursive function that would call itself when a sailor could not be saved, and reallocate different resources to the sailor before, and continue to do this until that sailor could be saved. This could also be extremely helpful in the case that we go above 100. However, once we received the third dataset, we found this to be unnecessary.

**Ethical Considerations**

We consider whether it is ethical to start treatment with the sailors that require the least amount of resources to recover, rather than starting treatment with the unhealthiest sailors. We consider our method more ethical when taking into account that we are maximizing the amount of lives saved.

One ethical consideration we make is in our ethical\_boosting function, that is, the function we make to give out our leftover resources on the third day. Originally, we just gave them out to any sailor whose health would not go above 100. However, this left big differences in the sailors’ health levels, some having very high T, while others were just barely above zero. To counteract this, we create the ethical\_boosting function to allocate the resources in order to give resources to the lowest T values.