# Week 2 Submission

**Question 1**

Problem: find a limit value *l* such that the cumulative existing booking values are maximal and less than *N.*

Input: a set of values corresponding to bookings, and the property's available time *N.*

Output: the adjusted set of bookings such that all values that were greater than *l* are now less than *l*.

**Test with 5 values, N = 100:**

20, 20, 20, 20, 20

Here we can set *l* to be the sum of the bookings/5. Easy, this is an ideal input.

30, 30, 30, 30, 30

Formula for sets of bookings with uniform values: sum of booking values/10 = y. Individual booking value/y = *l*.

15, 25, 35, 45, 55

Sum = 175, obviously 175 > 100. If we set *l* to 20, we get 15+20+20+20+20 = 95. Not bad but if we make *l* = 21 then we get right up to 99. This is optimal as it is the best value of *l* that doesn't breach the *N* and has a maximal value.

So in this:

- Sum/10 = 17.5, the only value that we don't need to change is less than this. So, let's see if we can tick 15 off as a value that doesn't need to be changed; N = 85, set of bookings is 25, 35, 45, 55. 85/4 = 21.25, take the int value and we have 21. *L* = 21
- So we have sum/10 = x, any values less than x don't need to be changed. Subtract the values smaller than x from N, then do N/number of values left = *l*. Make *l* an int so it rounds down.

25, 35, 15, 5, 90

- 5 and 15 are smaller than the sum/10 = 17. N – 5 – 15 = 80. 3 values left, 80/3 = 26.66 so *l* = 26. 5+15+26+26+26 = 98 < 100

What if there are 4 small values and 1 large one?

2, 2, 2, 2, 93.

- X = 10.1, 2's don't need to be changed, N = 92. One value left, so *l* = 92/1 = 92. 2+2+2+2+92 <= 100

**N = 69, values are 26, 33, 84, 4, 7**

- X = sum/10 = 15.4>4, 7. N – 4 – 7 = 58. 58/3 = 19.33, *l* = 19. 19+19+19+4+7 = 68 <= 69

```
makeTheValueLessThanTheLimitFunction(int bookings [], int N){
	If sum of bookings is less than N
		return l as the largest element in bookings
	Sum of the booking values divided by 10 = x
	M = values less than x subtracted from N
	Y = number of values greater than x
	//l needs to be an int so that it is rounded down to the nearest integer
	Return int l = M/Y
```

**Question 2**

a)

a) $n^2 + n + 1 = \Theta(n^3)$

$0 \le c_1 \times n^3 \le n^2 + n + 1 \le c_2 \times n^3$

choose $c_1 = 1$, $n = 2$, $c_2 = 1$

$0 \le 8 \le 9 \le 8$

For the given values, $9 \le 8$ is false, therefore $n^2 + n + 1 = \Theta(n^3)$ is disproven

b)

b) If $f_1(n) = O(g_1(n))$ and $f_2 = O(g_2(n))$, then $f_1(n) + f_2(n) = O(g_1(n) + g_2(n))$

Say $g_1(n) = n^2 + n$ and $g_2(n) = n^2 + 3n + 2$

$f_1(n) = O(g_1(n)) = O(n^2)$, $f_2(n) = O(g_2(n)) = O(n^2)$

$f_1(n) + f_2(n) = O(g_1(n) + g_2(n)) = O(n^2 + n^2) = O(2n^2) = O(n^2)$

c)

c) $\sum_{i=1}^{n} i = O(n^3)$

$\sum_{i=1}^{n} i = \frac{n(n-1)}{2} = \frac{n^2 - n}{2}$

$O\left(\frac{n^2 - n}{2}\right) = O(n^2) \neq O(n^3)$ $\therefore$ disproven