Question 1 (Proof of Correctness)

Proof of conquer f: (Merges the 2 individually sorted lists into a bigger overall sorted list

Basis:

$$conquer(l_1, []) = l1$$

Since l_1 is sorted & $l2$ is empty. ∴ ~~As appended~~ concatenated list will be sorted & $= l_1 1$

||ly $conquer([], l_2) = l_2$

∴ Basis is True

IH: Let l_1 & l_2 be 2 sorted lists & the function conquer $(l1, l2)$ returns a list $l3$ such that $l3$ is sorted and contains all elements of $l1$ & $l2$

[$l2 = x :: ls$ i.e a list headed by element x & followed by list ls]

IS: Let x_1 be an element appended to list $l1$ s.t. $x1 :: l1$ is sorted & we call conquer $(x1 :: l1, l2)$ = conquer $(x1 :: l1, x :: ls)$

∴ By the algo if $x_1 < x$, the funcⁿ returns $x1$ appended to list returned by conquer $(l_1, x :: ls)$

Now since $x_1 < x$ & both $l1$ & $l2$ are sorted lists ∴ $x1$ will be the smallest element in the list formed by concatenating 2 lists, marking its index as 0^{th}

Also, by IH we know that conquer $(l1, l2)$ is already sorted

∴ Funcⁿ returns the correct value

We can interchange $l1$ & $l2$ for proof of II cond^n

⊛ Correctness of Merge Sort

Basis: Consider ~~array~~ list of 1 element. Such array is already sorted.

Basis is correct

IH: Mergesort will sort any ~~array~~ list of length less than n

IS: Suppose we call func^n on list of size n

It will recursively call Mergesort of 2 lists of size n div 2 & $(n+1)$ div 2

By IH, these calls will sort the lists correctly

Now the func^n conquer is called, which as proved, will merge the 2 lists and sort them

Hence the given Algo is correct

Question 3 (Proof of Correctness)

(a) Correctness of intadd

Basis: intadd (l1, []) = l1.

This is basically equivalent to adding nothing to a number & ∴ returning same number

⇒ Basis holds true

IH: Let the function return correct value for 2 lists of sizes m & n respectively

IS: Let there be a I list of size m+1 & II of size n+1

Teacher's Signature.............

Date ______
Page ______

Let the units & tens places be x_1 & x_2 for I list & units place for II be $y1$

~~We already have~~

$\Rightarrow$ we need to perform :

$$\begin{array}{r} \_\_\_\_ \ \_\_ \quad x2 \quad x1 \\ + \ \_\_\_\_\_ \ \_\_ \quad y1 \\ \hline \end{array}$$

I Case

If $x1 + y1 > 9$ we will have 1 carried over to tens (by (1)) place, with $x1 + y1 - 10 = (x_1 + y_1) \ \text{mod} \ 10$ in ones place

$\hookrightarrow$ (i.e. $\Rightarrow x2 + 1 + y2$ where $y2$ is tens place for II list)

Since, by IH, we get correct evaluation for lists of size m & n

~~$\Rightarrow$ returning~~ ~~$\therefore$ given~~

II Case :

If $x1 + y1 \leq 9$ we will simply have to evaluate from tens place & since by IH ~~b~~ we have correct ans. for lists of size m and n respectively

$\Rightarrow$ Given Algorithm is correct

[$\because$ $(a_0 + 10a_1) + (b_0 + 10b_1) = (a_0 + b_0 - 10) + 10((a_1 + 1) + b_1)$] — (1)

(*) Proof of correctness for intsub

(For this, it is assumed that I integer > II because of instructor's response on Piazza)

Basis : intsub (l1, [ ]) = l1

This is equivalent to subtracting nothing from a number & hence returning the same ~~no~~ number

$\therefore$ ~~B~~ Basis is true

IH : Let the funcn return correct value for 2 lists of sizes m & n respectively

Teacher's Signature..............

IS: Let there be 2 lists of sizes m+1 & n+1 with units & tens place of I list: x1 & x2 and units & tens place of II list be y1 & y2

$\therefore$ We need to perform:

$$\begin{array}{r} \_\_\_\_\_ \; x2 \;\; x1 \\ - \;\_\_\_\_\_\; y2 \;\; y1 \\ \hline \end{array}$$

~~By III class algo~~

Case I: $x_1 - y_1 < 0$

By 3rd class algebra, we know that in such case 10 is added to the digit on top & tens digit of the number is reduced by 1

$(\because (a_0 + 10a_1) - (b_0 + 10b_1) = [(a_0+10) - b_0] + 10[(a_1-1) - b_1])$

Ones digit $= x_1 + 10 - y_1$ & $x_2 \to x_2 - 1$

Since by IH, we get correct value for list of size m & n

$\therefore$ Case I holds

Case II: $x_1 - y_1 > 0$

In this case, we simply perform algebra in same fashion from tens place

Now, since by IH, we get correct value for lists of sizes m & n

$\therefore$ Given algorithm is correct