This is Assignment 5 for MATH1005 students in a Friday workshop. It is due at 6 pm on the Thursday after Workshop 5 (6 days after was released).

There are four problems. The numbering of the problems is strange because the numbering is taken from a much larger document that has many problems from which I can select. As long as you can see four different problems, then you have the complete assignment.

You should write your best solutions to the problems here, and then upload your solutions before the due time. Here are three ways you may complete the assignment:

1. Print the assignment sheet. Write your solutions in pen or pencil on the print out. Scan your completed assignment, turn the file into a single .pdf file, then upload your solution file to Wattle.

2. Write your solutions in pen or pencil on blank paper. You should clearly label your solutions and you should write them in the order in which the problems appear in your assignment. Scan your completed assignment, turn the file into a single .pdf file, then upload your solution file to Wattle.

3. Download the assignment sheet to a tablet. Annotate the file using your favourite annotation software. **Flatten the file**–this makes your annotations a permanent part of the file, and if you do not do this then we see only a blank assignment in our grading software. Upload your flattened solution file to Wattle.

In all cases, the file you upload must be a .pdf file.

Please remember to plan your time carefully so you are not trying to submit your assignment at the last minute. No late work is accepted.

Please enjoy,

AP

Question 1#

(a) Evaluate: (i) $\displaystyle\sum_{k=0}^{5}(k-1)(k+1)$ (ii) $\displaystyle\prod_{k=0}^{5}\left(\frac{k-1}{k+1}\right)$

(b) The 'Catalan numbers' C_n, which have several applications in computing, may be defined implicitly by $C_0 = 1$ and $C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}$. Evaluate C_5.
See `https://en.wikipedia.org/wiki/Catalan_number` for many interesting properties of Catalan numbers, and of C_5 in particular.

Question 3^+

(a) Prove by mathematical induction that $\forall N \in \mathbb{N} \quad \displaystyle\sum_{n=1}^{N} n(n+1) = \frac{N(N+1)(N+2)}{3}$.

(b) [Challenge] Noting that $n(n+1) = n^2 + n$, use the formula of (a) to help evaluate the sum of the first one hundred squares; $1^2 + 2^2 + \cdots + 100^2$.
[You can use WolframAlpha to check your answer. Try "sum of n^2, n=1..100".]

Question 5* The letters of the word `KNIGHTED` form the list $(X_i)_{1..8} = (\texttt{K,N,I,G,H,T,E,D})$. This list is to be sorted into alphabetical order using Selection sort. The sorting is to be achieved by progressively modifying an index function π, rather than by shuffling members of the list itself. So initially

$$(X_i)_{1..8} = (X_{\pi(i)})_{1..8} \text{ where } \pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}$$

and when sorting is complete π is sufficiently changed so that $(X_{\pi(i)})_{1..8}$ is in order.

(a) First apply the Least Element algorithm to $(X_i)_{1..8}$. Demonstrate the application by completing the trace table at right.

i	2	3	4	5	6	7	8	9
m	1	1	3	4				
$x_{\pi(i)}$	N	I	G	H				-
$x_{\pi(m)}$	K	K	I	G				

(b) Write out the modified index function π resulting from (a).

(c) Now apply the Least Element algorithm to $(X_i)_{2..8}$ using this modified π, again demonstrating the application by a trace table.

(d) Write out the newly modified index function π resulting from (c).

(e) Without making trace tables, write out the state of index function π after each of the remaining applications of the Least element algorithm needed to complete the Selection sort of $(\texttt{K,N,I,G,H,T,E,D})$.

(f) What is the total number of comparisons used during this sort?

(g) By contrast, how many comparisons, in total, would be used to sort $(\texttt{K,N,I,G,H,T,E,D})$ using the Merge sort algorithm? To find out, carry out the Merge sort algorithm on $(\texttt{K,N,I,G,H,T,E,D})$ and carefully count the comparisons, remembering that when the Merge algorithm reaches a stage where one of its input lists is empty, it does not need any more comparisons to complete its task.

Question 6^{+} In lectures we saw how use the Merge sort algorithm to sort a sequence of length $n = 2^r$ into ascending order. In fact the algorithm can be applied to sequences of any length $n \in \mathbb{N}$. At each iteration the current sorted sub-sequences are merged in pairs as for the 2^r case but if there are an odd number of sub-sequences then the 'left over' one just joins, unchanged, the newly formed sub-sequences at the next iteration. This will mean that the merge algorithm will sometimes need to merge sequences of unequal lengths, but this causes no problems.

For example, if Merge sort is used to sort the letters of the word `PROVISIONAL` into alphabetical order then the subsequences at each stage will be:

after 0th iteration	`(P),(R),(O),(V),(I),(S),(I),(O),(N),(A),(L);`
after 1st iteration	`(P,R),(O,V),(I,S),(I,O),(A,N),(L);`
after 2nd iteration	`(O,P,R,V),(I,I,O,S),(A,L,N);`
after 3rd iteration	`(I,I,O,O,P,R,S,V),(A,L,N);`
after 4th iteration	`(A,I,I,L,N,O,O,P,R,S,V).`

(a) Apply the Merge sort algorithm to sort the letters of the word `SUBSTANTIATING` into alphabetical order, showing the results of each iteration as in the example above.

(b) How many comparison operations are used to merge sort `SUBSTANTIATING`? As in Worksheet Q5, remember that when the merge algorithm reaches the stage where one of its input lists is empty, it does not need any more comparisons to complete its task. For example, for `PROVISIONAL` there are only 5 comparisons during the first iteration, 8 in the 2nd, 7 in the 3rd and 5 in the last.

(c) How many comparison operations would be used if `SUBSTANTIATING` were sorted using the Selection sort algorithm?