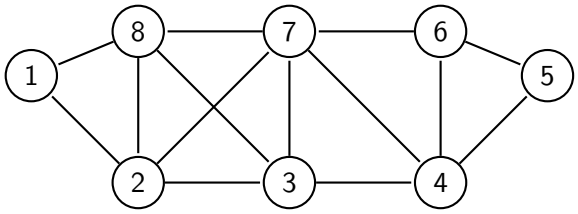
Hand in machine written solutions to problems 1, 2 and 3 by 3PM on Wednesday 21 March

Note: In each of the following problems you must describe the algorithm in your own words. You must *not* copy from lecture notes or elsewhere. You don't need to write pseudocode, just give an unambiguous description that your classmates should be able to use to perform the algorithm correctly by hand.



1. (a) Give an unambiguous and complete description of Dijkstra's algorithm. 3pts

   (b) Use this to find the minimum path lengths from node **1** to all other nodes in the graph above, where the edge between vertices i and j has weight given by the non-negative difference between the i-th and the j-th digit in your nine digit *Registration number*. For example, if your number is 201261834, then the weight of the edge between vertices 1 and 2 is $|2-0|=2$, and for the edge between vertices 3 and 7 it is $|1-8|=7$. 7pts

   Your Registration number begins with the year you started at Strathclyde

   Show, on a picture of the graph, *all the distances computed at each node*, during each step of the process, and circle the final distance for each one.

   Write a boxed list of all the distances computed from **1** to each of the other nodes, in increasing order of the labels of the nodes. Thus, since the distance from **1** to itself is 0 the list will begin $\boxed{0\ldots}$. *Make sure this list stands out in your solution.*

   Important instructions you must follow

2. (a) Give an unambiguous and complete description of the Bubblesort algorithm. 2pts

   (b) Input your nine digit registration number at a website you can find from the course webpage[1] to obtain a list of nine numbers. Sort this list by Bubblesort, showing the status of the list after each pass of Bubblesort through the entire list, on separate lines. At the *beginning of your solution* write your Registration number and list the number in the *fourth* place of the list after each pass. Here is an example of what the solution should look like, where I have written the fourth number in boldface: 3pts

   This number begins with 2015 if that's the year you started at Strathclyde

   Reg. no.: 201456789 4th letter: 7244

   3 1 6 **7** 2 4 5
   1 3 6 **2** 4 5 7
   1 3 2 **4** 5 6 7
   1 2 3 **4** 5 6 7

3. (a) Give an unambiguous and complete description of the Mergesort algorithm. 2pts

   (b) Use Mergesort to sort the list of 16 numbers obtained as in the previous problem. Display the results after each merging of all the pairs of sublists. At the *beginning of your solution* write your Registration number and list the last number in the first sublist after each step. A sample solution: 3pts

   Depending on how Mergesort is implemented, there may never be a program state where all the sublists have been merged as in the example, but you get the idea, I assume ...

   Reg. no.: 201456789 last number first block: 3,6,8,8

   3-6-8-4-1-7-5-2
   3**6**-48-17-25
   346**8**-1257
   1234567**8**

[1] https://personal.cis.strath.ac.uk/einar.steingrimsson/cs208/

Bonus problems on next page

BONUS PROBLEMS

Note: To get credit for a Bonus problem the solution must be very clear and convincing. You are welcome to discuss these with me, but you *may not* use any external resources to aid you, only your own thinking. Handing in a solution to a Bonus problem is a *declaration* that you have adhered to this. Credit for Bonus problems will be added to your mark, although not beyond 100% for the assignments in this part of the course.

You don't need to do the Bonus problems, but if you're looking for a challenge ...

4. Find a simple and elegant way to determine the number of passes required when sorting a list of n distinct numbers by Bubblesort. (In particular, the complexity of this should be linear at most.)

   The form of the solution could for example be along these lines (although this is certainly not a correct answer): The number of elements in the permutation that are immediately followed by a smaller element.)

   Here is one example of bubble sorting, pass by pass:

   561983274
   516832749
   156327489
   153264789
   132546789
   123456789

5. What is the average number of passes required when sorting a permutation of length n by Bubblesort?

   If you can't give an exact answer (I don't know it) for a general n, an asymptotic value when n tends to infinity would be valuable. Even good bounds (upper and/or lower) for this average or the asymptotic value would be interesting.

   One way to come up with a guess for an answer (especially if you think, as I do, that thinking hard can be painful) is to write a program that computes the number of passes, and check this for all permutations of short length and/or lots of randomly generated permutations of various lengths.

6. What is the average decrease in the number of inversions by a swap in Quicksort?

   Example: If you swap the 7 and the 4 in 7824 you undo the inversions 72, 74, 84, but introduce the new inversion 87, since the outcome is 4827, so the decrease is 2 inversions.