The University of Melbourne
COMP20007: Design of Algorithms
Semester 1 2021

Reading Time	15 minutes.
Writing Time	Three hours.

Instructions to Students

- This exam counts for 60% of your final grade.
- You should allow at least 15 minutes to scan and upload your solutions.
- Submissions uploaded past 30 minutes after the exam will not be accepted.
- The test is open book, which means you may only use course materials provided via the LMS or the text book but must not use any other resource including the Internet.
- You must not communicate with other students or make use of the Internet.
- Solutions must be written on separate pieces of paper with pen or pencil. You must write your solution to each question on a new sheet of paper and clearly indicate which question you are answering on each page.
- You must not use a tablet to complete this test.
- You must indicate which question is answered on which page via Gradescope.
- You must use a Scanner app on your smartphone to scan your solutions, email them to your laptop and then submit a PDF file to Gradescope via Canvas.
- A Gradescope guide to scanning your test can be found here:

  `https://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf`

Question 1 [6 marks]

(a) For the following pairs of functions, $f(n), g(n)$, determine if $f(n) \in \Theta(g(n)), f(n) \in O(g(n))$, or $f(n) \in \Omega(g(n))$, making the strongest statement possible. That is, if both $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$ are true, you **must** answer with the form $f(n) \in \Theta(g(n))$. You must answer with $f(n)$ on the left and $g(n)$ on the right, for example, you may not answer $g(n) \in O(f(n))$. You need to justify your answer.

i. $f(n) = n^2 log(n) + n(log(n))^2$ and $g(n) = nlog(n)$.

ii. $f(n) = \sum_{i=0}^{n} 2^i$ and $g(n) = 2^n$.

iii. $f(n) = 3^n$ and $g(n) = 3^{3n}$.

iv. $f(n) = (\sqrt{2})^{\log n}$ and $g(n) = \sqrt{n}$.

(b) Which of the following classes are not a subclass of $O(n^2)$?

1. $O(n)$
2. $\Omega(1)$
3. $\Theta(n \log n)$

(c) Describe the worst case running time of the following pseudocode function in Big-Oh notation in terms of the variable n. Make the strongest possible claim.

```
function WEIRD(n, sum)
    for i ← 0 to n × n − 1 do
        for j ← 0 to j − 1 do
            if (i % 10) == 0 then
                sum ← sum + 1
            j ← j + 1
        i ← i + 1
```

We know from the lectures that for $0 < \varepsilon < 1 < c$: $1 \prec \log n \prec n^{\varepsilon} \prec n^{c} \prec n^{\log n} \prec c^{n} \prec n^{n}$.

Remember that $\log n$ is an abbreviation for $\log_2 n$, the binary logarithm.

Remember also that for $a > 0$: $\sum_{i=0}^{n} (a^i) = \frac{1 - a^{n+1}}{1 - a}$.

We remind you also of the following conventions:

$$\sum_{i=m}^{n} (s_i) = s_m + s_{m+1} + \ldots + s_{n-1} + s_n \quad \text{and} \quad \prod_{i=m}^{n} (p_i) = p_m \cdot p_{m+1} \cdot \ldots \cdot p_{n-1} \cdot p_n.$$

Logarithm laws:

$$\log a + \log b = \log ab, \quad \log a - \log b = \log a/b \quad \text{and} \quad \log a^n = n \log a$$

.

Question 2 [4 marks]

(a) Solve the following recurrence relation given a constant $k > 0$:

$$T(n) = T(n-1) + k^n, \qquad T(0) = 1.$$

(b) Give a tight asymptotic upper bound for the following recurrence:

$$T(n) = 4T(n/8) + \sqrt[3]{n^2}$$

Master Theorem

For integer constants $a \geq 1$ and $b > 1$, and function f with $f(n) \in \Theta(n^d), d \geq 0$, the recurrence

$$T(n) = aT(n/b) + f(n)$$

(with $T(1) = c$) has solutions, and

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

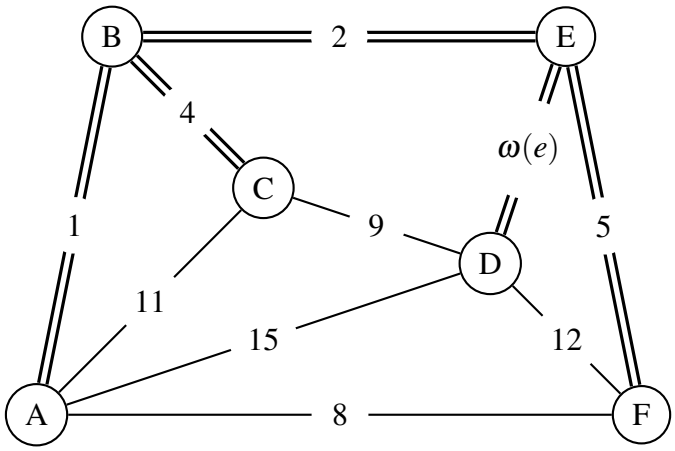
Note that we also allow a to be greater than b.

Question 3 [8 marks]

(a) What is the time complexity to compute the in-degree of every vertex in the graph (V,E) if the we use an adjacency list to represent the graph? Make the strongest statement possible.

(b) Let T be the minimum spanning tree of a graph G. Is it true that for any pair of vertices s and t the shortest path from s to t in T is the shortest path from s to t in G? If the answer is "yes", explain why; otherwise, give a counterexample.

(c) In the following graph all edges of the MST have been highlighted as double edges. What is the range of values for the weight $\omega(e)$ of edge DE?



(d) Assume that in the graph above the weight $\omega(e)$ of edge DE is 2. Step through Dijkstra's algorithm to calculate the single-source shortest paths from C to every other vertex. Show your steps in the table below as you have learnt it in the lectures where the left column are covered vertices so far and the right column lists a pair (distance, predecessor) if that node is still in the priority queue. Remember to use "nil" if the predecessor is not discovered or there is no predecessor.

Covered	*A*	*B*	*C*	*D*	*E*	*F*
–						
...	...	...	...	...	...	...

Question 4 [6 marks]

(a) Professor Breadth makes the following statement:

> Let CBT be a complete binary tree with n nodes. Finding a path from the root of CBT to a given vertex $v \in CBT$ using breadth-first search takes $\Omega(n)$ time in the worst case.

Explain in detail whether or not this statement is correct.

(b) Professor Union makes the following statement:

> If we have two sorted arrays, each of n real numbers and all numbers across the two arrays are distinct, then the construction a balanced binary search tree of the set $A_1 \cup A_2$ requires $\Omega(n \log n)$ time in the worst case.

Explain in detail whether or not this statement is correct.

Question 5 [12 marks]

A biomedical researcher needs to analyse gene sequences. Each sequence **S** is represented as a string with exactly $1,000$ characters over the alphabet $\{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$.

(a) Given a sequence **S**, the researcher needs to find the longest substring that consists of only two characters `A` and `T`. For example, if **S** was `"GATTGC"`, then `"ATT"` would be the answer. Develop an $O(n)$ algorithm to solve this problem. The algorithm should return the starting and ending positions of the substring (ie. return $(1,3)$ for the above example). If there are multiple such longest substrings, the pseudocode should return the one with the lowest starting position.

(b) Justify the time complexity for the algorithm you developed in Part **(a)**.

(c) Now the researcher wants to compress gene sequences. Describe the most space saving **fixed-size** encoding scheme for the above alphabet. Recall that in fixed-size encoding, the codeword of any character in the alphabet has the same length.

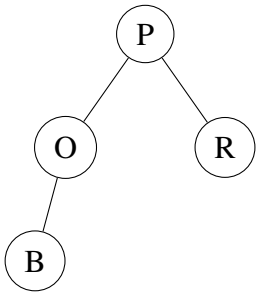
(d) For each sequence **S**, letters `A`, `C`, `G`, `T` appear with 500, 250, 125 and 125 times, respectively. Build a Huffmann tree for the alphabet, using the given frequencies of characters in **S**. Show your Huffmann tree and a possible assignment of codeword to each of the letters. Finally, show how many bits are saved when compressing **S** using this Huffman encoding, compared to the fixed-size encoding of Part **(c)** above.

(e) Now the researcher needs to *sort* the characters in a sequence **S**, returning anothter sequence with all characters sorted in alphabetical order. The algorithm needs to be as fast as possible while using small amounts of memory. Which algorithm do you recommend for this task? Justify your solution with no more than 2 sentences.

(f) Given that $\mathbf{S}_{sorted}$ is a sorted sequence obtained from Part **(e)**, the researcher now wants to find specific subsequences using Horspool's algorithm. Given the pattern `AAAC`, how many character comparsions are performed during the execution of the algorithm? Remember the string has exactly $1,000$ characters and for each sequence **S**, letters `A`, `C`, `G`, `T` appear with 500, 250, 125 and 125 times, respectively. Justify your answer.

Question 6 [9 marks]

(a) Suppose you have a dictionary implemented via an AVL-Tree. Here is the tree after inserting the letters `P, R, O, B`:



Insert the letter `L` into the tree above, draw the resulting tree and write down the balance factor for each node. Then, peform any necessary rotations to rebalance it and draw the resulting, balanced AVL-Tree. You should show each step necessary for rebalancing.

(b) Now assume you have the tree you obtained from Part **(a)** above. Insert the letter `E` into the tree and draw it with the balance factors for each node. Then, peform any necessary rotations to rebalance it and draw the resulting, balanced AVL-Tree. You should show each step necessary for rebalancing.

(c) Suppose now your dictionary is implemented via an 2-3 Tree. Insert the following letters in your 2-3 Tree, in this order (from left to right):

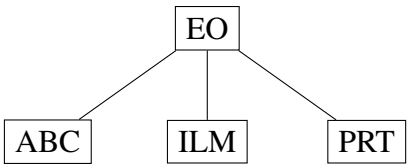
`P, R, O, B, L, E, M`

Draw the tree after each insertion step, including any required operations to keep the tree valid.

(d) Using the resulted 2-3 tree from Part **(c)** above, insert the letter "`A`" into the tree. Remember to include any necessary operations to keep the tree valid and show each step.

(e) Remember that in B-Tree of order m each non-root node has between $\lceil m/2 \rceil - 1$ and $m-1$ keys. Plus, for each internal node, its number of children is equal to the number of keys plus 1.

This is a B-Tree of order 5 after inserting letters `P, R, O, B, L, E, M, A, T, I, C`.



Delete the letter "`O`" from the B-tree above. Remember to perform any operations to ensure the B-tree is valid.

Question 7 [7 marks]

(a) The following array needs to be sorted in **descending order** (from the highest to the lowest number). Run the first iteration of SELECTIONSORT on the following array, assuming the array needs to be in descending order:

$$\left[\, 3,4,8,1,2,9,5 \,\right]$$

Show the state of the array after this operation. How many swaps were made?

(b) The following array needs to be sorted in **descending order** (from the highest to the lowest number). Show the state of the following array after performing a Hoare partition. Sort in descending order and choose the first element to be the pivot.

$$\left[\, 3,7,9,2,1,8,4,5 \,\right]$$

(c) Consider the following hybrid sorting algorithm which runs standard MERGESORT but when one of the subarrays reach size 10 or smaller, it calls INSERTIONSORT on the array:

```
function HybridSort(A[0...n-1])
    if n > 10 then
        B[0..⌊n/2⌋-1] ← A[0..⌊n/2⌋-1]
        C[0..⌊n/2⌋-1] ← A[⌊n/2⌋..n-1]
        HybridSort(B[0..⌊n/2⌋-1])
        HybridSort(C[0..⌊n/2⌋-1])
        Merge(B,C,A)
    else
        InsertionSort(A[0...m-1])
```

What is the worst case time complexity of HYBRIDSORT in terms of n (give your answer as a Big-Theta expression). You must provide a full derivation of the proof (you are allowed to use the Master Theorem if necessary).

Question 8 [8 marks]

A scout wants to raise funds by selling cookies in the neighbourhood's main street. The scout plans to visit houses in order and knows in advance how many cookies each house will buy. However, there is one caveat: **the scout cannot sell cookies in two consecutive houses**. The goal is to maximise the number of cookies sold in total. For example, if there are 6 houses in the street and the profit for each house is, in order:

$$[10, 20, 5, 15, 40, 10]$$

then, the maximum profit is obtained by selling cookies to houses 2 and 5 ($20 + 40 = 60$).

(a) You want to develop a Dynamic Programming (DP) solution for this problem. Define the base case for this problem.

(b) Define the recurrence relation for the DP solution.

(c) Develop the full DP algorithm and write it in pseudocode format. The algorithm should receive an array with the house profits and return the set of houses that maximises the total profit (in the example above, it would be houses 2 and 5).

END OF EXAM