The University of Melbourne
School of Computing and Information Systems
COMP90038 Algorithms and Complexity

Assignment 2, Semester 1 2018

Released: 7 May. Deadline: 21 May 9:00am

Objectives

To improve your understanding of data structures and algorithms for sorting and searching. To develop problem-solving and design skills. To improve written communication skills; in particular the ability to present algorithms clearly, precisely and unambiguously.

Problems

1. **[2 marks]**

   Consider the situation where you want to load n files of size $s_1, s_2, \ldots, s_n$ into a computer memory of size S. Unfortunately, it is not possible to simultaneously load all files into memory as $\sum_{i=1}^{n} s_i > S$. Therefore, you need to select a subset of the files to load such that $\sum_{i=1}^{k} s_i \leq S$ (where $k < n$).

   Assume that the files are sorted by non-decreasing size $s_i \leq \ldots \leq s_n$.

   (a) Does a greedy algorithm that selects files in order of non-decreasing s_i maximize the number of files stored in the memory? If you answer yes, provide a brief explanation. If you answer no, give a counter-example.

   (b) Does a greedy algorithm that selects files in order of non-decreasing s_i use as much of the memory capacity as possible? If you answer yes, provide a brief explanation. If you answer no, give a counter-example.

   Please keep your explanations concise – each part of the question is only worth 1 mark.

2. **[4 marks]**

   Consider two sets of integers, $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ where $m \leq n$.

   In this question, you must write two different versions of a CheckSubSet algorithm in pseudocode to test if X is a subset of Y. Each of your algorithms should return a value of True if X is a subset of Y, otherwise it returns False.

   You may call any of the algorithms introduced in the lectures (e.g. MaxElement($A[1..n]$)) to help you develop your solution. That is, you do not have to write the 'standard' algorithms – just use them. Therefore, you should be able to write each algorithm in approximately 8–12 lines of code.

   You must include appropriate comments in your pseudocode.

   (a) Write a pre-sorting based algorithm CheckSubSet(X, Y) to test if X is a subset of Y.

   (b) Now, write an $O(n)$ version of the CheckSubSet(X, Y) algorithm to test if X is a subset of Y. (*Hint:* how can hashing be used?)

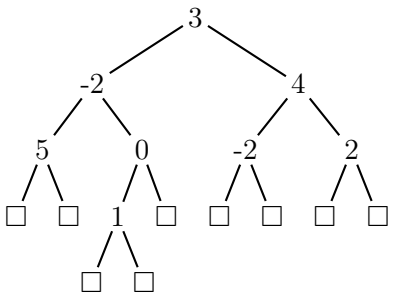
3. **[5 marks]**

Suppose T is a binary tree that stores an integer key value in each node. Assume the following notation/operations on a binary tree.

- the key $T.key$ is the root node's integer key value
- the left child $T.left$ is T's left subtree, which is possibly an empty tree (or *null*)
- the right child $T.right$ is T's right subtree, which is possibly an empty tree (or *null*)

(a) Write an efficient algorithm FindMaxProduct(T) in pseudocode that returns the maximum product of the key values on **all possible paths** in the tree T, which is passed as the input to the function.

For the tree below, your algorithm should return a value of 240 (from the path with key values $5 \times -2 \times 3 \times 4 \times -2$).



You must include appropriate comments in your pseudocode.

(b) Trace the output of your algorithm 'running' using the binary tree shown above. You may wish to include a table of node values and corresponding 'temp' values generated at each node as your algorithm proceeds.

4. **[4 marks]**

If we insert a set of n items into a binary search tree the resulting tree may be unbalanced. In the lectures, we examined how AVL trees can be used to solve this 'unbalanced' problem. In this question, you will investigate an alternative approach using a **treap**.

A treap is a binary search tree with a modified way of ordering the nodes with respect to the key values k_i and a min-heap with respect to the priority p_i.

Consider a set of records $R = \{r_1, r_2, \ldots r_n\}$. Each record r_i has a key k_i and independently of k_i a priority p_i. We will write a record as (k_i, p_i). The records are inserted into the treap such that the binary-search-tree property based on the record's key value is maintained and the priorities obey the min-heap order property.

You may assume that all key values and priorities are positive integers, and all priorities are distinct as well.

(a) Draw the treap T that contains the following seven record values:

$$\{(a, 9), (b, 3), (c, 7), (d, 2)(e, 6)(f, 5)(g, 4)\}.$$

(b) Write an algorithm BUILDTREAP(R) in pseudocode to construct a treap from the set of records R passed as an argument to your function. (*Hint*: use recursion) **You must include appropriate comments in your pseudocode**.

Your pseudocode should make reasonable assumptions about how trees are represented. The notation/syntax/style used in question 3 should also be adopted here. In addition, you may like to use:

- $T \leftarrow$ **new** $Tree$ to create a new tree;
- $T.key \leftarrow k$ and $T.priority \leftarrow p$ can be used to set the record attribute (field) integer values;
- *null* can be used to represent an empty tree

Submission and evaluation

- You must submit a PDF document via the LMS. Note: handwritten, scanned images, and/or Microsoft Word submissions are not acceptable — if you use Word, create a PDF version for submission.
- Marks are primarily allocated for correctness, but elegance of algorithms and how clearly you communicate your thinking will also be taken into account. The complexity of algorithms also matters, so you should think carefully about the time (and in some cases the space) complexity of any algorithms you write.
- We expect your work to be neat — parts of your submission that are difficult to read or decipher will be deemed incorrect. Make sure that you have enough time towards the end of the assignment to present your solutions carefully. Time you put in early will usually turn out to be more productive than a last-minute effort.
- You are reminded that your submission for this assignment is to be your own individual work. For many students, discussions with friends will form a natural part of the undertaking of the assignment work. However, it is still an individual task. You should not share your answers (even draft solutions) with other students. Do not post solutions (or even partial solutions) on social media. It is University policy that cheating by students in any form is not permitted, and that work submitted for assessment purposes must be the independent work of the student concerned.

  Please see https://academicintegrity.unimelb.edu.au

If you have any questions, you are welcome to post them on the LMS discussion board. You can also email the Head Tutor, Anagha Madhusudanan <`amadhusudana@student.unimelb.edu.au`> or the Lecturer, Michael Kirley <`mkirley@unimelb.edu.au`>. In your message, make sure you include COMP90038 in the subject header. In the body of your message, include a precise description of the problem.

Late submission will be possible, but **a late submission penalty will apply**: a flagfall of 2 marks, and then 1 mark per 12 hours late.

Extensions will only be awarded in extreme/emergency cases, assuming appropriate documentation is provided – simply submitting a medical certificate on the due date will not result in an extension.