University Number: _____

Solution

THE UNIVERSITY OF HONG KONG
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE

COMP2119C Introduction to Data Structures and Algorithms

Midterm Quiz

Date: Oct 24, 2024 Time allowed: 90 minutes

Write Your **University Number** in the designated space provided on every page.

INSTRUCTIONS:

1. Attempt ALL questions. The full mark for the quiz is 50 points.
2. Please write your answer in the designated space. If you run out of space, you can use the blank side of the previous page. Please indicate clearly when you are doing so.
3. log refers to log base 2.

University Number: _____

1. (a) (5 pts) Arrange the following FIVE functions in **increasing** order of asymptotic complexity.

   (A) $n^{1.5}$ (B) $n^{\log n}$ (C) $\log\log n + 10^{100}$ (D) $100000n$ (E) $500\log^2 n$

   Write the letters (A)-(E) in the space provided.

   Answer:
   (C) < (E) < (D) < (A) < (B)

   (b) (5 pts) Given $f(n) = n + \log n$ and $g(n) = \sqrt{n}$, **which one is true**, (i) $f(n) = O\big(g(n)\big)$ or (ii) $g(n) = O\big(f(n)\big)$? **Proof your answer.**

   Answer: $g(n) = O\big(f(n)\big)$.

   Proof: we want to prove $\exists\, c, n_0 > 0$, s.t., $0 \le \sqrt{n} \le c(n + \log n)$, $\forall n \ge n_0$.
   Now, $\sqrt{n} \ge 0, \forall n \ge 0$. Also,

$$\begin{aligned} \sqrt{n} &\le n, && \forall n \ge 1 \\ &\le n + \log n, && \forall n \ge 1 \end{aligned}$$

   Choosing $c = 1$, $n_0 = 1$ completes the proof.

University Number: _____

2. (a) (5 pts) Consider the following recursive function:

```
1 Function F(n):
2     if n = 1 then
3         return 1
4     else
5         return F(n − 1) + F(n − 1)
```

Let $T(n)$ be the **execution time** of $\mathrm{F}(n)$. Set up the recurrence formulation for $T(n)$, solve it and thereby find the time complexity (in big-O notation) of $\mathrm{F}(n)$.

Answer:

$$\begin{aligned} T(1) &= c_1 \\ T(n) &= 2T(n-1) + c_2, \qquad n > 1 \end{aligned}$$

for some constants c_1, c_2.
Solving the recurrence,

$$\begin{aligned} T(n) &= 2T(n-1) + c_2 \\ &= 2[2T(n-2) + c_2] + c_2 \\ &= 2^2 T(n-2) + (2+1)c_2 \\ &\vdots \\ &= 2^{n-1} T(n-(n-1)) + (2^{n-2} + \cdots + 1)c_2 \\ &= 2^{n-1} c_1 + (2^{n-1} - 1)c_2 \\ &= O(2^n) \end{aligned}$$

Note: Answer is accepted if $O(1)$ is used in place of c_1 and c_2. Marks will be deducted if 1 is used in place of c_1 and c_2.

University Number: _____

(b) (5 pts) Suppose we now replace line 5 in F(n) by

return 2 * F($n-1$)

Find the time complexity (in big-O notation) of F(n) after the change.

Answer:
Define $T(n)$ be the execution time of F(n) after the change.

$$\begin{aligned} T(1) &= c_1 \\ T(n) &= T(n-1) + c_2, \qquad n > 1 \end{aligned}$$

for some constants c_1, c_2.
Solving the recurrence,

$$\begin{aligned} T(n) &= T(n-1) + c_2 \\ &= [T(n-2) + c_2] + c_2 \\ &= T(n-2) + 2c_2 \\ &\vdots \\ &= T(n-(n-1)) + (n-1)c_2 \\ &= c_1 + (n-1)c_2 \\ &= O(n) \end{aligned}$$

Note: Answer is accepted if $O(1)$ is used in place of c_1 and c_2. Marks will be deducted if 1 is used in place of c_1 and c_2.

University Number: _____

3. (a) (10 pts) Consider a hash table T using the *open addressing*, with table size $m = 7$ and the primary hash function

$$h_1(k) = k \bmod m.$$

Assume that T is empty initially.

Show the result of **inserting the keys 33, 47, 23, 12** in the given order to T using the following collision resolution mechanism:

(i) quadratic probing: $h(k, i) = (h_1(k) + i^2) \bmod m$

Answer:

	0	1	2	3	4	5	6
T	12		23			33	47

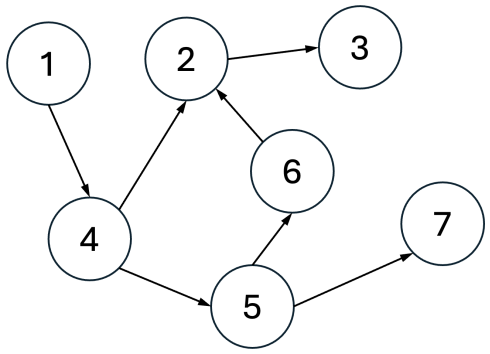
(ii) double hashing: $h(k, i) = \big(h_1(k) + i \times h_2(k)\big) \bmod m$ where $h_2(k) = k \bmod 5 + 1$.

Answer:

	0	1	2	3	4	5	6
T		47	23		12	33	

University Number: _____

(b) (5 pts) Consider the following directed graph with vertices labelled 1, 2, 3, 4, 5, 6, 7:



Give ALL possible vertex traversal orders of a Depth-First Search (DFS) of the graph starting from vertex 1.

Answer:

$1 \to 4 \to 2 \to 3 \to 5 \to 6 \to 7$

$1 \to 4 \to 2 \to 3 \to 5 \to 7 \to 6$

$1 \to 4 \to 5 \to 6 \to 2 \to 3 \to 7$

$1 \to 4 \to 5 \to 7 \to 6 \to 2 \to 3$

University Number: _____

4. Consider a string T given in the form of yZy' where y and y' are strings of equal length comprising characters 'a', 'b', 'c' only, and 'Z' is a special character at the middle of T. Some examples of T are 'abcaZabca', 'cbbZbbc', 'baaccZcabcc'.

   (a) (10 pts) Complete the algorithm IsPalindrom(T) that **makes use of a stack** to determine whether a given string T in the above form is a palindrome, i.e., T reads the same backward or forward. For example, 'cbbZbbc' is a palindrome.

   An empty stack S has been defined for you. You may use the following operations of a stack: **push(S, x)** which pushes x to S, **pop(S)** which pops the top element x from S, **empty(S)** which checks whether S is empty or not.

   Note: you may write the algorithm in pseudocode or in C/C++/Python languages. If you choose to write in C/C++/Python, make sure that you insert sufficient comments to help understand your algorithm.

   (b) (5 pts) What is the worst case time complexity of your algorithm? Explain briefly.

Answer:

```
Algorithm IsPalindrome(T):
    S ← init(S)
    scan through each character c in T from start to end
    while c != 'Z' do
        push(S, c)
    for each character c subsequent to 'Z' in T do
        x ← pop(S)
        if c != x then
            return False
    return True
```

Worst case complexity = $O(n)$.

The algorithm scans through each character in T once. For each character, it either pushes to or pops from the stack S, and each push/pop operation takes $O(1)$ time.

University Number: _____

- BLANK PAGE -
(You may continue your work for Q.4 here.)

- END OF PAPER -