Computer Science Qualifying Exam

AM Session

Practice Exam

Computer Science Department, Reed College

You have two hours to complete this portion of the exam. There are ve problems on 9 pages. Write your answers on the blank portions of each sheet, and on extra blank pages if needed. Make sure that your name and the problem number of your answer appear clearly on each sheet.

A M1.

Write your code for the `height` function here.

A M3. [12 points] Below is the Python de nition for a `MessageBoard`

Write your code for the `RentableMessageBoard` class on this page.

AM4. [12 points] Give the asymptotic worst-case running time for each of the following functions. Write this using Theta notation, fully simplified. (For example, $\Theta(n^2)$ or $\Theta(n \log n)$.) For each function, the input is a list and we define the size of the input list to be n.

(a) [3 points]

```
def functionA(list):
    a = 0
    for b in list:
        for c in list:
            if b + c >= 0:
                a = a + 1
            else:
                a = a - 1
    return a
```

(b) [3 points]

```
def functionB(list):
    g = 0
    for h in list:
        g = g + h
    i = 1
    j = 1
    while i < len(list):
        j = j * list[i]
        i = i * 2
    return g - j
```

(c) [3 points]

AM5. [12 points] Below are two Python classes that we use to de ne a doubly-linked list:

```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None
        self.prev = None

class LinkedList:
    def __init__(self):
        self.first = None

    def prepend(self, value):
        newNode = Node(value)
        newNode.next = self.first
        if self.first is not None:
            self.first.prev = newNode
        self.first = newNode

    def output(self):
        current = self.first
        while current is not None:
            print(current.value)
            current = current.next
```

In the code's use below, we build a linked list with 712 stored both at positions 0 and 1, 8 stored at position 2, 18 stored at position 3, and 37 stored at position 4.

```
>>> ll = LinkedList()
>>> ll.prepend(37)
>>> ll.prepend(18)
>>> ll.prepend(8)
>>> ll.prepend(712)
>>> ll.prepend(712)
>>> ll.output()
712
712
8
18
37
```

Write a method reverse that modi es the list so that the values are in the opposite order they started in. For full credit, you should not modify the value of any Node.

Write your code for the `reverse` function on this page.

Computer Science Qualifying Exam

PM Session

Practice Exam

Computer Science Department, Reed College

You have two hours to complete this portion of the exam. There are ve problems on 7 pages. Write your answers on the blank portions of each sheet, and on extra blank pages if needed. Make sure that your name and the problem number of your answer appear clearly on each sheet.

Complete each problem to the best of your ability. When the problem requests that you devise a program or a fragment of a program's code in a certain programming language you should do your best to produce working code in that language { syntactically correct and runnable { that meets the speci cation of the problem.

When you are providing code answers, if you you forget some aspect of the language, you can communicate your intended meaning for partial credit. As an example, if you forget how logical conjunction works in

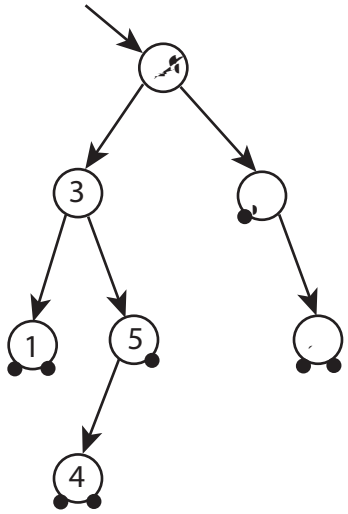
PM1. [12 points] Write a C++ function `chooseBit`. This is a function that takes as arguments an unsigned 8-bit integer value (of type `uint8_t`) and an integer between 0 and 7 and returns the bit at the specified position in the 8-bit value. The return value should be either a 1 or a 0. For example, if the bits of the

PM2. [12 points] Write a Python function `greater`. This is a function that returns another function. It should take an integer as its argument. It should return a function that takes an integer argument and returns whether or not the argument is greater than the original.

```
>>> x = greater(12)
>>> x(20)
True
```

PM4. [12 points] The C++ code for binary search tree node is given below left and a picture of a binary search tree is shown to the right:

```
struct node {
    uint32_t data;
    struct node* parent;
    struct node* left_child;
    struct node* right_child;
};
```



Write a C++ fuction `insert` for inserting a value into the binary search tree. The function should take as an argument a pointer to the root node of the tree and a value to add into the tree. It should create a new node containing the value and add it into the input tree as a leaf in the appropriate location according to the binary search tree property. For this problem, we will make two assumptions. The first is that the input tree will never be empty. The second is that the tree is not allowed to contain duplicates. Preventing duplicates means that if the value already exists in the tree, the function should not create a new node and should return `false`. If the value does not exist, a new node should be created and the function should return `true`.

PM5. [12 points] Using only the MIPS32 instructions on the guide in the bottom half of this page, write MIPS assembly code that determines whether a given value can be found in the elements pointed to by an array of pointers. Remember that pointers in MIPS are 32 bits.

Preconditions: Your code can assume that

- Elements in the array are valid 32-bit pointers that each point to a 32 bit unsigned integer.
- Register `$a0` gives the value being searched for (a 32 bit unsigned integer).
- Register `$a1` gives the (word-aligned) address of the first element in the array.
- Register `$a2` gives the length of the array.

Postconditions: Upon completion

- Register `$v0` contains the index of the first element that points to a value matching the input, or -1 otherwise.

Having completed its work, your code should jump to a label `done`. You can use any of the registers `$t0` through `$t9` to perform your calculations.

Please write your MIPS code on the next page, using only instructions from those listed below.

MIPS32 Assembly Guide

`li` $RD, *value*	loads an immediate value into a register
`lw` $RD, ($RS)	loads a register from memory at an address specified in a register
`sw` $RS, ($RD)	stores a register into memory at an address specified in a register
`addu` $RD, $RS1, $RS2	add two registers, storing the sum in another
`subu` $RD, $RS1, $RS2	subtract two registers, storing the difference in another
`addiu` $RD, $RS1, *value*	add a value to a register, storing the sum in another
`move` $RD, $RS	copy a register's value to another
`j` *label*	jump to a labeled line
`blt` $RS1, $RS2, *label*	jump to a labeled line if one register's value is less than another
`bltz` $RS, *label*	jump to a labeled line if a register's value is less than zero
`gt`, `le`, `ge`, `eq`, `ne`	other conditions than `lt`
`sll` $RD, $RS, *value*	shifts the value in the source register *value* bits left
`sllv` $RD, $RS1, $RS2	shifts the value in a source register another source register bits left

Write your MIPS code here:

Computer Science Qualifying Exam
Discrete Mathematics (DM) Session

10am - 12pm, March 5, 2023

Computer Science Department, Reed College

You have two hours to complete this portion of the exam. There are four problems on 6 pages. Write your answers on the blank portions of each sheet, and on extra blank pages if needed. Make sure that your

DM1. [4 points each, 24 total] Below are several short questions. Write your answer clearly in the box provided. We have provided extra space if you need to do work, but there is no partial credit for these problems | if you know the answer, there's no need to write out a long explanation. When it asks for a number, please give a speci c number. (For example, don't write \5!", write \120". You have a

(d) A casino o ers a simple dice game. A player picks a number between 1 and 4, then rolls two (fair) 4-sided dice. If one die rolls the number the player guessed, they win a dollar. If two dice roll that number, they win two dollars. If neither die does, then they lose a dollar. Is it smart to play this game? What is the average gain/loss for the player?

(e) How many numbers in $f1; 2; 3; : : : ; 1999g$ are integer multiples of 3 or 4 but not 12?

(f) Four numbers, a, b, c, and d are chosen at random from the set $f0; 1; 2; 3; : : : ; 1999g$. (The numbers are chosen uniformly and independently. There is no requirement that they be distinct.) What is the probability that $ab \quad cd$ will be an even number?

DM2. [12 points] Prove that for any integers a and b we have

$$\gcd(a^5; b^5) = \gcd(a; b)^5$$

where gcd denotes the greatest common divisor. (Hint: One good solution involves prime factorization.)

DM3. [12 points] Let p be a prime number greater than 5. Show that p^2 1 must be divisible by 24. (Hint: Consider the values of p^2 mod 3 and mod 8.)

DM4. [12 points] Let $S = \{2, 3, 5, 7, 11, 13, 17, 19\}$ be the set of prime numbers less than 20. For a subset $A \subseteq S$, define the sum of S to be $\text{sum}(A) := \sum_{k \in A} k$. Prove that there are four nonempty subsets of S with the same sum. (Hint: what is the largest possible $\text{sum}(A)$ for $A \subseteq S$? What is the minimum $\text{sum}(A)$?)