Homework 1

Dez Wolken

September 2024

1 Problem 1

Describe the standard algorithm for finding the hexadecimal representation of a positive decimal integer.
In English:
Divide the integer part by 16 and keep going until you can't. Then, combine the remainders starting from most recent. For the numbers after the decimal point, multiply by 16 and take note of the integer part of the result. Keep multiplying the fractional part by 16 until you can no longer. Once done, combine the result from the first part, add a decimal point, then include the combined result from the second part.
In pseudocode:

```
// store hexadecimal number// hexnum = ['0'] times 100
//set var for loop// i = 0
while input isn't 0:
    //set variable for remainder// remain = 0
    remain = input/16
    if remain is less than 10:
        hexnum[i] = character(remain +48)
        i = i+1
    otherwise:
        hexnum[i] = character(remain +55)
        i=i+1
    n = int(input/16)
//outputs the results starting from most recent//
j = i-1
while j isn't 0:
print hexnum[j]
j = j-1
```

2 Problem 2

Merging two sorted lists of numbers into a single sorted list.
Lists represented as arrays:
Because they're already sorted, you can do a for loop of the second array (j) in the for loop for the first array (i). If i is less than j, then you add it to the new combined list, and keep going until all elements that fit are added.
Lists represented as linked lists:
Create a new head pointer to the new linked list for the combination of the two. Check to see which node is smaller of the two linked lists, then append it to the new list.

3 Problem 3

$$S(n) = 1^3 + 2^3 + ... + n^3$$

Set up and solve a recurrence relation for the number of times the algorithm's basic operation is executed: 2n
How does this algorithm compare with the straightforward nonrecursive algorithm for computing this sum?
The total number of times this algorithm's basic operation executes non recursively is n, which causes the non recursive algorithm to be faster than its recursive counterpart.