PRACTICE PROBLEMS FOR THE SECOND MIDTERM
CSCI 417

Review old homework problems. In addition, do the following problems.

Problem 1. Suppose a divide-and-conquer algorithm works by subdividing a problem into 3 subproblems of size $n/3$. Suppose that the combining step takes n^2 operations. Write a recurrence relation to describe this situation. Assuming that the time required for a problem of size 1 is 1 time unit, to what big-oh class does this algorithm belong?

Problem 2. An algorithm takes 0.5 ms for input size 100. Approximate how long will it take for input size 500 if the running time is:

(1) linear
(2) $n \log(n)$
(3) quadratic
(4) cubic

Problem 3. Use Mater Theorem to find the asymtopic size of the following function.

(1) $T(n) = 2T(\frac{n}{2}) + n^2$.
(2) $T(n) = 4T(n/3) + n$.
(3) $T(n) = 4T(n/4) + n^3$.

Problem 4. We are given a list of coin values, denoted by `coin_list`, and a target value N . We have studied the minimum coin change problem. In this problem, we make a small adjustment: we want to find the minimum number of coins required to achieve a total value of N under the following additional condition:

(1) The first coin in `coin_list` has a limited availability defined by a constant C.
(2) All other coin values in the list are available in unlimited quantities.

The input of this problem is a a tuple containing the list of coin values, the target value, and the limit for the first coin:

$$(\texttt{coin\_list}, N, C)$$

Let us consider the following two examples.

(1) Input:

$$([1, 2, 4], 10, 2)$$

- Coin values are $1, 2, 4$.
- The total amount required is 10.
- The coin value 1 has a maximum availability of 2.
- The solution is 3 because $10 = 4 + 4 + 2$.

(2) Input:

$$([4, 2, 1], 8, 1)$$

- Coin values are $1, 2, 4$.
- The total amount required is 8.
- The coin value 4 can be used only once.
- The solution is 4 because $8 = 4 + 2 + 2 + 2$.

How would you adjust your algorithm to compute the minimum number of coins with the additional constrain explained above? Explain your solution with the following examples

(1) ([2, 5, 7], 14, 1).
(2) ([7, 5, 2], 14, 1).

Problem 5. Write a recursive function to determine if a string is a palindrome.

Problem 6. Write a recursive function to count the number of 1 in the binary representation of a number n. For example, if $n = 15$ then the answer should be 4 since $15 = \overline{1111}$ in base 2.