

CSCI 135 Software Design and Analysis I

Homework 7

Due 12/12/08

Saad Mneimneh
Visiting Professor
Hunter College of CUNY

Problem 1: Repeated squaring

In class we saw a time efficient algorithm to compute b^n where $n \geq 0$ is an integer. The algorithm is based on repeated squaring and is given by the following **recursive definition** of b^n :

$$b^n = \begin{cases} 1 & n = 0 \\ b \cdot b^{n-1} & n \text{ odd} \\ (b^{n/2})^2 & n \text{ even} \end{cases}$$

A direct translation to C++ gives the following **recursive implementation**:

```
float exp(float b, int n) {
    if (n>0) {
        if (n%2==1)
            return b*exp(b,n-1);
        else
            return exp(b,n/2)^2;
    }
    else
        return 1;
}
```

- Rewrite the recursive definition for b^n by exchanging the $n/2$ and 2 exponents of the third case (when n is even).
- Provide an recursive implementation for the modified definition.
- Based on the modified definition, describe an **iterative process** for computing b^n .
- Based on the iterative process, write a **tail recursive implementation** for computing b^n and transform it to an **iterative implementation**.
- What is the advantage of this final implementation?

Problem 2: Two-dimensional arrays with vectors

(a) Write a program to:

- Ask the user to input two integers m and n .
- Declare a two-dimensional array of booleans of size $m \times n$. You must declare this dynamically using the **new** operator because m and n are not known at compile time (remember to free memory at the end of the program).
- Fill the array randomly with 0s and 1s.

(b) Repeat part (a) using C++ vectors. In this case you will be using a vector of vector of booleans. Enjoy the fact that there is no need to dynamically allocate memory in this case.

(c) Write a function that takes the two-dimensional array as a parameter (so you should know how to pass the array as argument and what type the parameter(s) should be) and returns the number of adjacent 1s in it.

(d) Repeat (c) using C++ vectors. What is the best way to pass the vector to the function?

Problem 3: string vs. char *

(a) Using C++ strings, write a function that accepts a string s as a parameter, and returns a string t which is the reverse of s .

(b) Comment on the issues involved if char * is used instead of string. What features about the string class helps avoid those issues?