0 Math Exercises

Problem 0.1. DeMorgan's laws state that

$$\neg(P \vee Q) \iff \neg P \wedge \neg Q \tag{1}$$

$$\neg(P \wedge Q) \iff \neg P \vee \neg Q \tag{2}$$

Prove one of these identities by way of a truth table.

Problem 0.2. Consider the following recursively defined function,

$$f(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ n \cdot f(n/2) & \text{if } n \mod 2 = 0 \\ n + f(n-1) & \text{if } n \mod 2 = 1 \end{cases}$$

Calculate $f(20)$.

Problem 0.3. Let a, b be characters, ϵ be the empty string, and $\circ$ denote string concatenation. Consider the following definition of a foo,

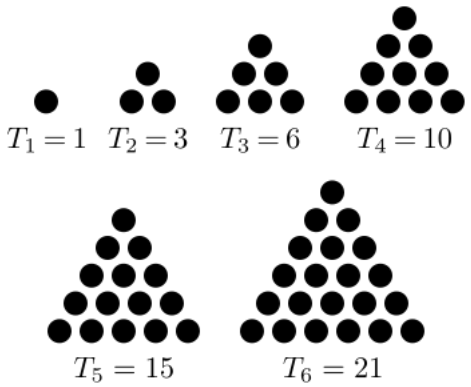
$$foo = \begin{cases} \epsilon \\ a \circ foo \circ a \\ a \circ foo \circ b \\ b \circ foo \circ a \\ b \circ foo \circ b \end{cases}$$

(a) Is *aba* a foo?

(b) Is *babb* a foo?

(c) In more intuitive terms, what is a foo?

Problem 0.4. A *Triangular Number* counts objects that are arranged into equilateral triangles (see the image below, lifted from Wikipedia).



(a) Come up with a recursive definition for T_n, the n^{th} triangular number.

(b) Come up with a non-recursive formula for T_n (extra credit: prove this formula via induction).

(c) Let S_n denote the sequence of non-zero perfect squares, i.e. $S_1 = 1, S_2 = 4, S_3 = 9, \ldots$. Prove that $S_n = T_n + T_{n-1}$ for $n \geq 2$. *Hint:* the closed formula from part (b) may be helpful.It may also help to try devising a "proof by picture" first.

1 Computer Problems

All problems from this section should be uploaded to `Github`

Problem 1.1. Write a C program that takes in two integers and outputs the division algorithm performed with those two numbers as inputs. Below is an example of output.

```
bash-3.2$ gcc divisionAlg.c -o divisionAlg
bash-3.2$ ./divisionAlg
Please enter a number: 6
Please enter another number: 2
6 = 3 * 2 + 0
bash-3.2$ ./divisionAlg
Please enter a number: 89
Please enter another number: 7
89 = 12 * 7 + 5
bash-3.2$ ./divisionAlg
Please enter a number: -56
Please enter another number: 3
-56 = -18 * 3 + -2
bash-3.2$ ./divisionAlg
Please enter a number: 20
Please enter another number: -2
20 = -10 * -2 + 0
bash-3.2$
```

Problem 1.2 (Quadratic Formula)**.** Write a `C` program that takes as input three real numbers a, b, c which determine a quadratic equation of the form

$$ax^2 + bx + c = 0,$$

and outputs any real roots of that equation. Your program should determine whether your equation has zero, one, or two real roots, and indicate this to the user (you do not need to compute complex roots).