Algorithms: Solutions 1

```
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                                 X
                              X  X
                              X  X
                              X  X
                              X  X
number of                     X  X
homeworks                  X  X  X
                        X  X  X  X
                        X  X  X  X
                        X  X  X  X
                  X  X  X  X  X  X
            X  X  X  X  X  X  X  X
   --------------------------------
   0  1  2  3  4  5  6  7  8  9  10
                 grades
```

The histogram shows the distribution of grades, from 0 to 10.

Problem 1
Let $A[1..n]$ be an array of n distinct numbers. Write an algorithm INVERSIONS(A, n) that determines the number of inversions in $A[1..n]$.

INVERSIONS(A, n)
counter $\leftarrow 0$
for $i \leftarrow 1$ **to** $n - 1$
    do for $j \leftarrow i + 1$ **to** n
        do if $A[i] > A[j]$
            then *counter* $\leftarrow$ *counter* $+1$
return *counter*

The time complexity of the INVERSIONS algorithm is $\Theta(n^2)$.

Problem 2

Let $A[1..n]$ be a sorted array of n distinct numbers. Write an efficient algorithm BINARY-SEARCH(A, n, k) that finds a given value k in the array $A[1..n]$. It should return the index of the found element; if the array does not include k, the algorithm should return 0.

```
BINARY-SEARCH(A, n, k)
p ← 1
r ← n
while p < r
    do q = ⌊(p + r)/2⌋
        if k ≤ A[q]
            then r ← q
            else p ← q + 1
if k = A[p]
    then return p
    else return 0
```

The time complexity of this binary search is $\Theta(\lg n)$.

Problem 3

(a) $1 + x + x^2 + x^3 + ... + x^n = \frac{x^{n+1}-1}{x-1}$ (where $x \neq 1$).

We use a proof by induction. Clearly, the equality holds for $n = 1$; we now show that, if it holds for n, then it also holds for $n + 1$:

$$\begin{aligned} 1 + x + ... + x^n + x^{n+1} &= \frac{x^{n+1} - 1}{x - 1} + x^{n+1} \\ &= \frac{x^{n+1} - 1 + x^{n+1} \cdot (x - 1)}{x - 1} \\ &= \frac{x^{n+1} - 1 + x^{n+2} - x^{n+1}}{x - 1} \\ &= \frac{x^{(n+1)+1} - 1}{x - 1} \end{aligned}$$

(b) $(1 + 2 + 3 + 4 + ... + n)^2 = 1^3 + 2^3 + 3^3 + 4^3 + ... + n^3$.

We observe that the equality holds for $n = 1$, and apply induction to "step" from n to $n+1$:

$$\begin{aligned} (1 + 2 + ... + n + (n + 1))^2 &= (1 + 2 + ... + n)^2 + 2 \cdot (1 + 2 + ... + n) \cdot (n + 1) + (n + 1)^2 \\ &= 1^3 + 2^3 + ... + n^3 + 2 \cdot \frac{n \cdot (n + 1)}{2} \cdot (n + 1) + (n + 1)^2 \\ &= 1^3 + 2^3 + ... + n^3 + n \cdot (n + 1)^2 + (n + 1)^2 \\ &= 1^3 + 2^3 + ... + n^3 + (n + 1) \cdot (n + 1)^2 \\ &= 1^3 + 2^3 + ... + n^3 + (n + 1)^3 \end{aligned}$$