Analysis of Algorithms: Results of Exam 2

```
                                      X
                                      X
                                      X
                                      X
                                      X
                                      X
number              X                 X
of exams            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     X     X     X     X     X     X     X
      0-50 51-55 56-60 61-65 66-70 71-75 76-80 81-85 86-90 91-95 96-100
```

The histogram shows the distribution of grades, *not* including the bonus.

Problem 9
Write an algorithm that reverses all edges of a given graph, that is, replaces every edge (u, v) with the opposite edge (v, u), and returns the resulting graph of reversed edges.

We denote the adjacency list of a vertex u in the initial graph G_1 by $\textit{Adj-List}_1[u]$, and the adjacency list of the same vertex in the reversed graph G_2 by $\textit{Adj-List}_2[u]$. The following algorithm constructs G_2 in $O(V + E)$ time:

Reverse-Edges(G_1)
for each $u \in V[G_1]$
    do initialize an empty list $\textit{Adj-List}_2[u]$
for each $u \in V[G_1]$
    do for each $v \in \textit{Adj-List}_1[u]$
            do add u to $\textit{Adj-List}_2[v]$
return G_2

Problem 10
Suppose that you are using a programming language that supports four operations on real numbers: addition, subtraction, multiplication, and division; the running time of each operation is constant, that is, $\Theta(1)$. Write an efficient algorithm Polynomial(x, A, n) for computing the value of a polynomial. The arguments of the algorithm are a value of x and an array of coefficients $A[0..n]$, and the output is the value of the following polynomial:

$$A[n] \cdot x^n + A[n-1] \cdot x^{n-1} + A[n-2] \cdot x^{n-2} + ... + A[1] \cdot x + A[0].$$

The computation takes $\Theta(n)$ time:

Polynomial(x, A, n)
power $\leftarrow 1$
sum $\leftarrow A[0]$
for $i \leftarrow 1$ **to** n
    do *power* $\leftarrow$ *power* $\cdot\, x$ $\triangleright$ It equals x^i.
        sum $\leftarrow$ *sum* $+ A[i] \cdot$ *power* $\triangleright$ It is $A[i] \cdot x^i + ... + A[0]$.
return *sum*