Exercise for Analysis of Algorithms

Exercise 10

Solve the following recurrence: Let $a_0 = 1$, $a_1 = 1$, $a_2 = 4$ and

$$a_n = 2a_{n-1} - a_{n-2} + 2a_{n-3}, \text{ for } n \geq 3.$$

Solution:

The characteristic polynomial is $z^3 - 2z^2 + z - 2$. This can be factorized as $(z^2+1)(z-2)$, which gives the set of roots as $\{\pm i, 2\}$. The general solution is thus:

$$a_n = \alpha \cdot 2^n + \beta \cdot (-i)^n + \gamma \cdot i^n.$$

Substituting the values for $n = 0, 1, 2$, we obtain the following system of linear equations:

$$\alpha + \beta + \gamma = 1 \tag{1}$$
$$2\alpha - i\beta + i\gamma = 1 \tag{2}$$
$$4\alpha - \beta - \gamma = 4 \tag{3}$$

Solving this system yields $\alpha = 1$, $\beta = -i/2$ and $\gamma = i/2$. Note that β and γ are complex conjugates. The solution is thus:

$$\begin{aligned} a_n &= 2^n - \frac{i}{2} \cdot (-i)^n + \frac{i}{2} i^n \qquad (4) \\ &= 2^n + (1 + (-1)^{n+1}) \cdot \frac{i^{n+1}}{2}. \qquad (5) \end{aligned}$$

This solution can be written in a much nicer form using Euler's formula:

$$e^{i\theta} = \cos\theta + i \sin\theta.$$

We may write $i^{n+1}/2$ as follows:

$$\begin{aligned} \frac{i^{n+1}}{2} &= \frac{1}{2}\left(\cos\frac{\pi}{2} + i \sin\frac{\pi}{2}\right)^{n+1} \\ &= \frac{e^{\frac{i\pi}{2}\cdot(n+1)}}{2}. \end{aligned}$$

Similarly, $(-1)^{n+1} i^{n+1}/2$ may be written as $\frac{1}{2} \cdot e^{-\frac{i\pi}{2}\cdot(n+1)}$. Now note that

$$\frac{e^{\frac{i\pi}{2}\cdot(n+1)} + e^{-\frac{i\pi}{2}\cdot(n+1)}}{2} = \cos\left(\frac{\pi(n+1)}{2}\right).$$

Thus $a_n = 2^n + \cos\frac{\pi(n+1)}{2}$.

Exercise 11
Solve the following recurrence: Let $a_0 = 0$, $a_1 = 3$ and

$$a_n = 4a_{n-1} - 4a_{n-2} \text{ for } n > 1.$$

Solution:
The characteristic polynomial is $x^2 - 4x + 4 = (x-2)^2$ with the only root $x_0 = 2$. The solution is therefore of form $a_n = \lambda 2^n + \mu n 2^n$. Our constraints yield $\lambda = 0$ and $\mu = \frac{3}{2}$.

Exercise 12
Consider the following program: The input to this program is an array $a[0, \ldots, n-1]$

```
int sel_sort ( int a[], int n ) {
  for ( int i = 0; i < n; ++i ) {
    int min = i;
    for ( int j = i; j < n; ++j ) {
      if ( a[j] < a[min] ) {
        min = j;
      }
    }
    int temp = a[i];
    a[i] = a[min];
    a[min] = temp;
  }
}
```

that contains n pairwise distinct integer keys in random order.

a) Explain how this program sorts the given array.

b) Analyse how often each instruction of the program is executed on average depending on n.

c) There is only one instruction whose analysis is not trivial. Which one is it?

   Make a table for small values of n by hand that lists the results for this instruction. Compare the table entries with the results from your closed formula that you obtained in b).

Solution:
The algorithm is, as indicated by the name, selection sort: it finds the minimal element of the yet-to-be-sorted part and appends it to the sorted part by exchange.
For part b), the outer loop is execute n times, therefore each statement not in the inner loop is executed that often. The if-statement is executed exactly $\frac{n(n+1)}{2}$ times, which leaves the min $= j$ statement.
The first comparison ($j = i$ to i) always fails, therefore the expected number of executions (cf. Problem 1-2) then is

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} \frac{1}{j-i+1} = \sum_{i=0}^{n-1} \sum_{j=1}^{n-i-1} \frac{1}{j+1} = \sum_{i=0}^{n-1} (H_{n-i} - 1) = (n+1)H_n - 2n.$$

Exercise 13
Solve the following recurrence and find a nice representation of the solution (in a mathematical sense).

$$\begin{aligned} c_0 &= 2 \\ c_1 &= 4 \\ c_n &= c_{n-2}^{\log c_{n-1}} \end{aligned}$$

Hint: Let F_n be the nth Fibonacci number. Write c_n as some function of F_n.

Solution:
We apply the logarithm and obtain

$$\log c_n = \log c_{n-1} \cdot \log c_{n-2}.$$

In order to obtain a sum instead of a product, we repeat this and obtain

$$\log \log c_n = \log \log c_{n-1} + \log \log c_{n-2}.$$

Substituting $d_n = \log \log c_n$ yields

$$\begin{aligned} d_0 &= 0 \\ d_1 &= 1 \\ d_n &= d_{n-1} + d_{n-2} \end{aligned}$$

Since this describes the Fibonacci numbers, we obtain $c_n = 2^{2^{F_n}}$, where F_n denotes the n-th Fibonacci number.