Problem 1. We have a recursive solution to Hanoi tower problem with 3 pegs:

```cpp
void Hanoi3(peg A, peg B, peg C, int n)
{ if (n == 0) return;
    Hanoi3(A, C, B, n-1);
    move a disc from A to B;
    Hanoi3(C, B, A, n-1);
}
```

and the exact count how many disc movements are made: \( T_3(n) = 2^n - 1 \).

a) Assuming that each move takes 1 second, what is the number of years needed to execute \( Hanoi3(0,1,2,64) \)? Round the answer to the nearest power of 2, e.g. “approximately \( 2^{20} \) years”.

Answer. A year has 365 \times 24 \times 3600 = 31,536,000 \approx 33,554,432 = 2^{25} \) seconds. Thus moving 64 discs with 3 pegs requires approximately \( 2^{64}/2^{25} = 2^{39} \) years.

Now suppose that we have 4 pegs, and otherwise the same rules. We may have a function \( f(n) \) that allows to formulate the following recursive algorithm:

```cpp
void Hanoi4(peg A, peg B, peg C, peg D, int n)
{ if (n < 3) {
    Hanoi3(A, B, C, n);
    return;
}
    Hanoi4(A, D, B, C, n-f(n));
    Hanoi3(A, B, C, f(n));
    Hanoi4(D, B, A, C, n-f(n));
}
```

b) Show that if for every \( n > 2 \) we have \( 0 < f(n) < n \) then this is a correct algorithm.

Answer. If we have \( n < 3 \) discs, we use \( Hanoi3 \), so we can rely on the correctness of \( Hanoi3 \). For a larger \( n \) we will perform the reasoning of the inductive step.

Let \( N \) denote the top \( n - f(n) \) discs and \( F \) denote the bottom \( f(n) \) discs.

We start with configuration \( A(NF), B(), C(), D() \), meaning, \( N \) and \( F \) on peg \( A \), nothing on other pegs.

- \( Hanoi4(A, D, B, C, n-f(n)) \) changes the configuration to \( A(F), B(), C(), D(N) \).
- \( Hanoi3(A, B, C, f(n)) \) changes the configuration to \( A(), B(F), C(), D(N) \). Note that pegs \( B \) and \( C \) were empty, so \( Hanoi3 \) can operate without obstruction.
- \( Hanoi4(D, B, A, C, n-f(n)) \) changes the configuration to \( A(), B(NF), C(), D() \). Note that discs of \( F \) that were initially on \( B \) are larger than the discs of \( N \), so they do not obstruct the movement of \( N \) when we execute \( Hanoi4(D, B, A, C, n-f(n)) \).

The last configuration is the desired one.
c) For $2 < n \leq 20$, find values for $f(n)$ that result in the minimum number of moves by Hano14.

**Answer.** We fill a table that gives the number of moves that results from different choices of $f$, the best choices will be possible values for $f(n)$:

<table>
<thead>
<tr>
<th>$n$</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2T_4(n-1) + T_3(1)$</td>
<td>7</td>
<td>11</td>
<td>19</td>
<td>27</td>
<td>35</td>
<td>51</td>
<td>67</td>
<td>83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2T_4(n-2) + T_3(2)$</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>21</td>
<td>29</td>
<td>37</td>
<td>53</td>
<td>69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2T_4(n-3) + T_3(3)$</td>
<td>NA</td>
<td>NA</td>
<td>17</td>
<td>25</td>
<td>33</td>
<td>41</td>
<td>57</td>
<td>73</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>$2T_4(n-4) + T_3(4)$</td>
<td>NA</td>
<td>NA</td>
<td>17</td>
<td>21</td>
<td>25</td>
<td>33</td>
<td>41</td>
<td>49</td>
<td>65</td>
<td>81</td>
</tr>
<tr>
<td>$2T_4(n-5) + T_3(5)$</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>33</td>
<td>37</td>
<td>41</td>
<td>49</td>
<td>57</td>
<td>65</td>
<td>81</td>
</tr>
<tr>
<td>best $f(n)$</td>
<td>2</td>
<td>2,3</td>
<td>2,3</td>
<td>3</td>
<td>3,4</td>
<td>3,4</td>
<td>3,4</td>
<td>4</td>
<td>4,5</td>
<td>4,5</td>
</tr>
</tbody>
</table>

Such calculations give the best values for $f(n)$:

<table>
<thead>
<tr>
<th>$n$</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(n)$</td>
<td>4,5</td>
<td>4,5</td>
<td>5</td>
<td>6,6</td>
<td>5,6</td>
<td>5,6</td>
<td>5,6</td>
<td>5,6</td>
<td>6</td>
</tr>
<tr>
<td>$T_4(n)$</td>
<td>97</td>
<td>113</td>
<td>129</td>
<td>161</td>
<td>193</td>
<td>225</td>
<td>257</td>
<td>289</td>
<td>321</td>
</tr>
</tbody>
</table>

**d)** Suggest a formula for $f(n)$ that results in a low (as low as possible) number of moves needed to move 64 discs with 4 pegs.

**Answer.** For every $n$ up to 21, an optimum $f(n)$ equals $\sqrt{2n}$. Applying it leads to $T_4(64) = 18,433$.

**e)** Find the order of growth of the logarithm of the number of moves that results from your formula. Usually, people use notation like $2^{\Theta(f(n))}$.

**Answer.** Assume that $T_4(n) \leq C^{\sqrt{n}}$.

Calculus workout: $\sqrt{n} - \sqrt{n-1} \geq k\sqrt{n} - \sqrt{n} - 1 = \frac{k}{2\sqrt{n}}$. Thus $\sqrt{n} - \sqrt{n-\sqrt{2n}} > 0.7$.

We will prove that $T_4(n) \leq C^{\sqrt{n}}$ for some $C$.

$$T_4(n) = 2T_4(n-\sqrt{2n}) + 2^{\sqrt{2n}} \leq 2C^{-0.7}C^{\sqrt{2n}} + 2^{\sqrt{2n}} \leq T(n)$$

This calculation is valid if $2^{\sqrt{2n}} < 0.5C^{\sqrt{n}}$ and $C^{-0.7} < 0.25$, and this is true for a sufficiently large $C$.

**Problem 5, page 128.**

a. $a = 4$, $b = 2$, $c = 1$, $A = 4/2 > 1$, leaf dominated, $n^{log_2 4} = n^2$.

b. $a = 4$, $b = 2$, $c = 2$, $A = 4/2^2 = 1$, balanced, $n^2 \log n$.

c. $a = 4$, $b = 2$, $c = 3$, $A = 4/2^3 < 1$, root dominated, $n^3$. 