**Problem 1:** For each pseudocode below, give the number of letters printed as a function of $n$, using the Θ-notation. For the first three programs give a recurrence and its solution. For the last two programs, give the solution and a brief justification (at most 20 words).

<table>
<thead>
<tr>
<th>pseudocode</th>
<th>Solution and recurrence or justification</th>
</tr>
</thead>
</table>
| **procedure** PrintAs($n$)  
  if $n > 1$ then  
    print("A")  
    PrintAs($n/3$)  
  $A(n) = A(n/3) + 1$  
  $A(n) = \Theta(\log n)$ |
| **procedure** PrintBs($n$)  
  if $n > 1$ then  
    for $j \leftarrow 1$ to $4n$  
      do print("B")  
    PrintBs($n/3$)  
    PrintBs($n/3$)  
  $B(n) = 2B(n/3) + 4n$  
  $B(n) = \Theta(n)$ |
| **procedure** PrintCs($n$)  
  if $n > 1$ then  
    for $j \leftarrow 1$ to $n^2$  
      do print("C")  
    for $i \leftarrow 1$ to $5$ do  
      PrintCs($n/2$)  
  $C(n) = 5C(n/2) + n^2$  
  $C(n) = \Theta(n^{\log 5})$ |
| **procedure** PrintDs($n$)  
  for $j \leftarrow 1$ to $n$ do  
    $k \leftarrow 1$  
    while $k < n$ do  
      print("D")  
      $k \leftarrow 2k$  
  $D(n) = \Theta(n \log n)$  
  internal loop makes $\Theta(\log n)$ iterations  
  because $k$ doubles at each step |
| **procedure** PrintEs($n$)  
  for $i \leftarrow 1$ to $n^2$ do  
    for $j \leftarrow 1$ to $2n$ do  
      print("E")  
  $E(n) = \Theta(n^3)$  
  for each of $n^2$ iterations of external loop  
  internal loop makes $2n$ iterations |
Problem 2: (a) Explain how the RSA cryptosystem works by filling in the table below.

<table>
<thead>
<tr>
<th>Initialization</th>
<th>Determine ( p, q, ) and ( n ):</th>
<th>( p, q ) are different primes and ( n = pq )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula for ( \phi(n) ):</td>
<td>( \phi(n) = (p - 1)(q - 1) )</td>
<td></td>
</tr>
<tr>
<td>Determine ( e ) and ( d ):</td>
<td>( e ) can be any number between 1 and ( n ) that is relatively prime to ( \phi(n) ), and ( d = e^{-1} \pmod{\phi(n)} )</td>
<td></td>
</tr>
<tr>
<td>Public and secret keys:</td>
<td>( P = (n, e), S = d )</td>
<td></td>
</tr>
</tbody>
</table>

Encryption: \( E(M) = M^e \pmod{n} \)  
Decryption: \( D(C) = C^d \pmod{n} \)

(b) Below you are given five choices of parameters \( p, q, e, d \) of RSA. For each choice tell whether these parameters are correct\(^1\) (write YES/NO). If yes, give an encoding of \( M = 3 \). If not, give a brief justification (at most 10 words).

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( e )</th>
<th>( d )</th>
<th>correct?</th>
<th>justify if not correct / encode ( M = 3 ) if correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>Y</td>
<td>Computing modulo 35: ( 3^5 = 243 = 33 )</td>
</tr>
<tr>
<td>11</td>
<td>27</td>
<td>13</td>
<td>55</td>
<td>N</td>
<td>27 is not a prime</td>
</tr>
<tr>
<td>17</td>
<td>5</td>
<td>5</td>
<td>13</td>
<td>Y</td>
<td>Computing modulo 85: ( 3^5 = 243 = 73 )</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>3</td>
<td>67</td>
<td>N</td>
<td>( p ) and ( q ) cannot be equal</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>5</td>
<td>27</td>
<td>N</td>
<td>( 5^{-1} \neq 27 \pmod{60} )</td>
</tr>
</tbody>
</table>

\(^1\)To clarify, correctness refers to whether these parameters satisfy the conditions in the algorithm.
Problem 3: (a) Give a complete statement of Fermat’s Little Theorem.

Theorem: Let \( p \) be a prime number and \( a \in \{1, 2, ..., p - 1\} \). Then \( a^{p-1} \equiv 1 \pmod{p} \).

(b) Use Fermat’s Little Theorem to compute the following values. In the second example, show your work.

\[ 35^{130} \mod 131 = 1 \]

\[ 3^{14074} \mod 71 = 10 \]

Computing modulo 71: \( 3^{14074} = 3^{14070} \cdot 3^4 = 1 \cdot 81 = 10 \).
**Problem 4:** For each $n \geq 0$ we define a binary tree $T_n$ as follows. $T_0$ is a single node and $T_1$ is also a single node. For $n \geq 2$, $T_n$ is obtained by creating two new nodes and adding copies of $T_{n-1}$ and $T_{n-2}$ as their subtrees, as in the picture below on the left:

![Binary tree diagram]

The picture on the right shows tree $T_3$ (with subtrees $T_2$ and $T_1$ marked).

Let $A_n$ be the number of leaves in $T_n$. (For example, $A_0 = A_1 = 1$, $A_2 = 3$ and $A_3 = 7$, as can be seen in the picture above.) Give a formula for $A_n$. You need to show your work, all steps. First, give a recurrence equation with a brief justification. Then solve this recurrence. At each step explain what you are computing.

The recurrence is

$$A_n = 2A_{n-1} + A_{n-2} \quad \text{for } n \geq 2$$

$$A_0 = 1$$

$$A_1 = 1$$

Justification for the recurrence: the leaves of $T_n$ are either the leaves of two subtrees $T_{n-1}$ or one subtree $T_{n-2}$.

The characteristic equation is $x^2 - 2x - 1 = 0$. The roots are $1 + \sqrt{2}$ and $1 - \sqrt{2}$. So the general solution is

$$A_n = \alpha_1 (1 + \sqrt{2})^n + \alpha_2 (1 - \sqrt{2})^n.$$  

Using the initial conditions, we get equations:

$$\alpha_1 + \alpha_2 = 1$$

$$\alpha_1 (1 + \sqrt{2}) + \alpha_2 (1 - \sqrt{2}) = 1$$

The solution is $\alpha_1 = \alpha_2 = \frac{1}{2}$. So the final solution is

$$A_n = \frac{1}{2}(1 + \sqrt{2})^n + \frac{1}{2}(1 - \sqrt{2})^n.$$
Problem 5: The Duggars are about to buy t-shirts for their 19 children, one for each. They need
- at least 2 blue t-shirts,
- at least 5 red t-shirts,
- at least 1 pink t-shirt, and
- at least 2 and not more than 10 yellow t-shirts.

How many different choices of t-shirt colors satisfy these requirements?

The answer is the number of non-negative integral solutions of
\[ b + r + p + y = 19 \]
\[ 2 \leq b \]
\[ 5 \leq r \]
\[ 1 \leq p \]
\[ 2 \leq y \leq 10 \]

After eliminating lower bounds (by substitutions), this reduces to computing the number of non-negative integral solutions of
\[ b + r + p + y = 9 \]
\[ y \leq 8 \]

Let \( S \) be the number of all non-negative integral solutions and \( S(P) \) the number of non-negative integral solutions that satisfy condition \( P \). Then
\[ S(y \leq 8) = S - S(y \geq 9) = \binom{12}{3} - \binom{3}{3} = 220 - 1 = 119. \]

So the answer is 119.
Problem 6: (a) Give Euler’s inequality for planar graphs, and use it to show that the graph below is not planar.

Euler’s inequality: In a planar graph with \( n \geq 3 \) vertices the number of edges \( m \) satisfies \( m \leq 3n - 6 \).
In this graph we have \( n = 7 \) and \( m = 16 \). These numbers do not satisfy Euler’s inequality, so \( G \) is not planar.

(b) Determine which of the following two graphs are planar. Justify your answer and show your work.

Graph \( H \) is planar. The picture below on the left shows a planar drawing of \( H \). Graph \( G \) is not planar, because it contains a sub-division of \( K_5 \), shown below on the right.
Problem 7: Use induction to prove that \( \sum_{k=1}^{n} k^3 = \frac{1}{4}n^2(n+1)^2 \) for all integers \( n \geq 1 \).

**Base case:** For \( n = 1 \), the left-hand side is \( \sum_{k=1}^{1} k^3 = 1 \) and the right-hand side is \( \frac{1}{4}1^2(1+1)^2 = 1 \) as well.

**Inductive step:** Assume that \( \sum_{k=1}^{n} k^3 = \frac{1}{4}n^2(n+1)^2 \). We want to show that this equation holds for the next value of \( n \), that is \( \sum_{k=1}^{n+1} k^3 = \frac{1}{4}(n+1)^2(n+2)^2 \). Starting from the left-hand side, and using the inductive assumption, we proceed as follows:

\[
\sum_{k=1}^{n+1} k^3 = \sum_{k=1}^{n} k^3 + (n+1)^3 \\
= \frac{1}{4}n^2(n+1)^2 + (n+1)^3 \\
= \frac{1}{4}(n+1)^2[n^2 + 4(n+1)] \\
= \frac{1}{4}(n+1)^2(n+2)^2.
\]

as needed.
Problem 8: We have a set of $2n$ players in a chess tournament, where $n \geq 1$. Let $f(n)$ be the number of ways to divide them into pairs for the first round of the tournament. Prove that

$$f(n) = \frac{(2n)!}{2^n n!}.$$  

For example, consider the case when $n = 2$, that is have four players. Let's call them A, B, C, D. There are three possible pairings: (AB, CD), (AC, BD), and (AD, BC). This agrees with the formula, because $f(2) = (2 \cdot 2)!/(2^2 \cdot 2!) = 4!/4 = 3$.

*Hint:* One way to approach this is to derive a recurrence equation for $f(n)$ and then prove that the above formula is its solution. Another way is to show a relation between pairings and permutations of the players.

**Solution 1:** For $n = 1$ we have two players and one pairing, so $f(1) = 1$. Consider some $n > 1$. The last player can be paired with any of the other $2n - 1$ players. Once we choose the pairing for the last player, the remaining players can be paired in $f(n - 1)$ ways. Thus we have the recurrence

$$f(1) = 1$$

$$f(n) = (2n - 1) f(n - 1)$$

It remains to verify that the formula above satisfies this recurrence. Indeed:

$$(2n - 1) \cdot f(n - 1) = (2n - 1) \cdot \frac{(2(n - 1))!}{2^{n-1}(n - 1)!} = \frac{(2n - 1)(2n - 2)!}{2^{n-1}(n - 1)!} = \frac{2n(2n - 1)(2n - 2)!}{2^nn!} = \frac{(2n)!}{2^nn!} = f(n),$$

as claimed.

**Solution 2:** Consider any of the $(2n)!$ permutations of the players, say $x_1, x_2, \ldots, x_{2n}$. This permutation defines a pairing where each odd-numbered player is paired with the next player: $x_1x_2, x_3x_4, \ldots, x_{2n-1}x_{2n}$. However, each pairing can be obtained in many ways from this construction: in each pair the two players can be exchanged in two ways, for the total of $2^n$ ways, and the $n$ pairs themselves can be obtained in any order, and there are $n!$ such orders. Therefore the number of pairings will be $(2n)!$ divided by $2^n n!$, which is exactly our formula.

**Solution 3:** Let’s try brute force: pick the pairs one by one. The first pair can be selected in $\binom{2n}{2} = 2n(2n-1)/2$ ways. Once we choose this pair, the second pair can be chosen in $(2n-2)(2n-3)/2$ ways, and so on. This will give us

$$\frac{2n(2n - 1)(2n - 2)...1}{2^n} = \frac{(2n)!}{2^n}$$

ways to choose the pairings. However, the $n$ pairs in each pairing can be selected in all possible orderings, and there are $n!$ such orderings. Thus we need to divide the above value by $n!$, which gives us $f(n) = (2n)!/(2^nn!)$. 

CS111 FINAL A SOLUTIONS, Spring’15, June 6.