| **CS 506 Online** | **Homework 10** | Prof. David Nassimi |
|---|---|---|
| Foundations of Comp Sci | Number Theory and Cryptography | |

Study Module 11: Number Theory and Cryptography

1. (a) List all prime numbers between 2 and 100. Use the fact that a composite integer $n$ must have a prime factor $\leq \sqrt{n}$ to limit the search. Thus, a number $< 100$ is prime if and only if it is not a multiple of $\{2, 3, 5, 7\}$. Other useful facts:
   - An integer is divisible by 3 if and only if sum of its digits are divisible by 3.
   - An integer is divisible by 5 if and only if it ends with a 5.

   (b) How many prime numbers are there in your list in the range 2 to 100?

**Solution:**

(a) 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97.
(b) How many? 25

2. (a) Find the Greatest-Common-Divisor (GCD) of the following pair of numbers by finding their prime factorizations: $a = 7623,\ b = 7425$.
   (b) Use Euclid's algorithm to find GCD of the above integers $a, b$. Use the iterative algorithm. Also show the computation of integers $(s, t)$ so that $\gcd(a, b) = sa + tb$.

**Solution:**
(a)

$$a = 7623 = 3^2 * 7 * 11^2;$$
$$b = 7425 = 3^3 * 5^2 * 11;$$
$$GCD = 3^2 * 11 = 99$$

(b)

| $a$ | $b$ | $q = \lfloor a/b \rfloor$ | $r = a - qb$ $= a \bmod b$ |
|---|---|---|---|
| 7623 (1,0) | 7425 (0,1) | 1 | 198 (1,-1) |
| 7425 (0,1) | 198 (1,-1) | 37 | 99 (-37,38) |
| 198 (1,-1) | 99 (-37,38) | 2 | 0 |
| 99 (-37,38) | 0 | | |

$$\gcd(7623{,}7425) = 99 = -37a + 38b$$
$$s = -37, t = 38$$

3. (a) Find inverse of integers 1 to 10 mod 11. Tabulate the results. You may find the values by inspection.
(b) Find inverse of integers 1 to 13 mod 14, if they exist. Tabulate the results.

Solution:
(a)

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $i^{-1}$ $mod$ 11 | 1 | 6 | 4 | 3 | 9 | 2 | 8 | 7 | 5 | 10 |

(b)

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i^{-1} mod$ 14 | 1 | - | 5 | - | 3 | - | - | - | 11 | - | 9 | - | 13 |

4. Use the iterative GCD algorithm to compute the inverse of 37 mod 121.

Solution:

| $a$ | $b$ | $q = \lfloor a/b \rfloor$ | $r = a \bmod b$ |
|---|---|---|---|
| 121 $(1, 0)$ | 37 $(0, 1)$ | 3 | $10 = 121 - 3 * 37$ $(1, -3)$ |
| 37 $(0, 1)$ | 10 $(1, -3)$ | 3 | $7 = 37 - 3 * 10$ $(-3, 10)$ |
| 10 $(1, -3)$ | 7 $(-3, 10)$ | 1 | $3 = 10 - 7$ $(4, -13)$ |
| 7 $(-3, 10)$ | 3 $(4, -13)$ | 2 | $1 = 7 - 2 * 3$ $(-11, 36)$ |

Therefore, $\gcd(121, 37) = 1 = -11 * 121 + 36 * 37$.
So, $37^{-1}\ mod\ 121 = 36$.

Note: If the latter number was a negative number (instead of +36), then we would compute mod 121 to get a positive result.

5. Use exponentiation method to find each inverse. Recall $x^{-1} mod\ n = x^{\phi(n)-1} mod\ n$.
(a) Inverse of 5 mod 17
(b) Inverse of 5 mod 21

Solution:
(a) Inverse of 5 mod 17. Let $x = 5,\ n = 17$. Since $n$ is prime, $\phi(n) = n - 1$.

$$x^{-1} = x^{\phi(n)-1} mod\ n = 5^{15} mod\ 17 = 7$$

(a) Inverse of 5 mod 21. Let $x = 5,\ n = 21 = 3 * 7,\ \ \phi(n) = (3 - 1)(7 - 1) = 12$.

$$x^{-1} = x^{\phi(n)-1} mod\ n = 5^{11}\ mod\ 21 = 17.$$

6. Let $n = 14 = 2 * 7$.
   (a) Compute and tabulate $5i \; mod \; n$ for $i = 1, 2, \cdots, \; n - 1$. Observe the result is a permutation of $(1, 2, \cdots, n - 1)$, as proved by Lemma 6.

   (b) Compute and tabulate $6i \; mod \; n$ for $i = 1, 2, \cdots, \; n - 1$. Explain why the result is not a permutation of $(1, 2, \cdots, n - 1)$.

   (c) Let $Z_{14}^*$ be set of positive integers smaller than 14 and prime relative to 14.
   For each $x \in Z_{14}^*$, compute and tabulate a row $xi \; mod \; 14, i \in Z_{14}^*$.
   Observe that every row is a permutation of $Z_{14}^*$, as proved by Lemma 8.

Solution:
(a)

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $5i \; mod \; 14$ | 5 | 10 | 1 | 6 | 11 | 2 | 7 | 12 | 3 | 8 | 13 | 4 | 9 |

(b)

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $6i \; mod \; 14$ | 6 | 12 | 4 | 10 | 2 | 8 | 0 | 6 | 12 | 4 | 10 | 2 | 8 |

The result is not a permutation of $(1,2, \cdots, 13)$ because $\gcd(6,14) \neq 1$.
Lemma 6 required gcd=1.

(c)

| $i \in Z_{14}^*$ | 1 | 3 | 5 | 9 | 11 | 13 |
|---|---|---|---|---|---|---|
| $3i \; mod \; 14$ | 3 | 9 | 1 | 13 | 5 | 11 |
| $5i \; mod \; 14$ | 5 | 1 | 11 | 3 | 13 | 9 |
| $9i \; mod \; 14$ | 9 | 13 | 3 | 11 | 1 | 5 |
| $11i \; mod \; 14$ | 11 | 5 | 13 | 1 | 9 | 3 |
| $13i \; mod \; 14$ | 13 | 11 | 9 | 5 | 3 | 1 |

For every $x \in Z_{14}^*$, the row $xi \; mod \; 14$ is a permutation of $Z_{14}^* = \{1, 3, 5, 9, 11, 13\}$, as proved by Lemma 8.

7. Let $p = 7$. For each $x = 2, \; 3, \cdots, p - 1$ compute and tabulate a row $(x^i \bmod p)$ for $i = 1, 2, \cdots, p - 1$.
   (a) Relate the results to Fermat's Little Theorem.
   (b) Which column gives the inverse, $x^{-1} \bmod p$? Explain.

**Solution:**

| $i$ | 1 | 2 | 3 | 4 | $5 = p - 2$ | $6 = p - 1$ |
|---|---|---|---|---|---|---|
| $2^i \ mod\ 7$ | 2 | 4 | 1 | 2 | 4 | 1 |
| $3^i \ mod\ 7$ | 3 | 2 | 6 | 4 | 5 | 1 |
| $4^i \ mod\ 7$ | 4 | 2 | 1 | 4 | 2 | 1 |
| $5^i \ mod\ 7$ | 5 | 4 | 6 | 2 | 3 | 1 |
| $6^i \ mod\ 7$ | 6 | 1 | 6 | 1 | 6 | 1 |

(a) Column $6 = p - 1$ demonstrates Fermat's Little Theorem.
That is, $x^{p-1} \ mod\ p = 1$.
(b) Column $5 = p - 2$ gives inverse of each $x$ value (highlighted in yellow). That is, since $p$ is prime, $\phi(p) = p - 1$, and $x^{p-2} = x^{\phi(p)-1} = x^{-1} \quad (mod\ p)$.

8. Let $n = 10 = 2 * 5$. Thus, $\phi(10) = (2-1)(5-1) = 4$, $Z_{10}^* = \{1, 3, 7, 9\}$.
For each $x \in Z_{10}^*$ compute and tabulate a row $(x^i \bmod n)$ for $i = 1, 2, 3, 4, 5$.
(a) Which column relates to Euler's Theorem? Explain
(b) Which Column relates to Corollary 1? Explain
(c) Which column gives the inverse, $x^{-1} \bmod n$? Explain.

Solution:

| $i$ | 1 | 2 | 3 | $4 = \phi(n)$ | 5 |
|---|---|---|---|---|---|
| $1^i mod\ 10$ | 1 | 1 | 1 | 1 | 1 |
| $3^i mod\ 10$ | 3 | 9 | 7 | 1 | 3 |
| $7^i mod\ 10$ | 7 | 9 | 3 | 1 | 7 |
| $9^i mod\ 10$ | 9 | 1 | 9 | 1 | 9 |

(a) Euler's Theorem says that for every $x \in Z_n^*$, $x^{\phi(n)} \ mod\ n = 1$. Here, $\phi(n) = 4$, and column 4 verifies this fact.
(b) From column $5 = \phi(n) + 1$, we see the value equals $x$, as proved by Corollary 1.
(c) Column $3 = \phi(n) - 1$ produces $x^{-1} \ mod\ n$.
In summary, columns (3, 4, 5) respectively produce $(x^{-1},\ 1,\ x)$.

9. Let $n = 10 = 2 * 5$. So, $\phi(n) = (2-1)(5-1) = 4$. For each positive integer $x < n$, compute and tabulate a row $(x^i \bmod n\ )$ for $i = 1$ to 5.
(a) Explain how the results in column 5 relate to Theorem 6.
(b) Explain how the values in column 4 relate to Euler's Theorem.
(c) Explain why the value in column 4 cannot be 1 for any $x \notin Z_n^*$.

Solution:

| $i$ | 1 | 2 | 3 | 4 | $5 = \phi(n) + 1$ |
|---|---|---|---|---|---|
| $1^i mod\ 10$ | 1 | 1 | 1 | 1 | 1 |
| $2^i mod\ 10$ | 2 | 4 | 8 | 6 | 2 |
| $3^i mod\ 10$ | 3 | 9 | 7 | 1 | 3 |
| $4^i mod\ 10$ | 4 | 6 | 4 | 6 | 4 |
| $5^i mod\ 10$ | 5 | 5 | 5 | 5 | 5 |
| $6^i mod\ 10$ | 6 | 6 | 6 | 6 | 6 |
| $7^i mod\ 10$ | 7 | 9 | 3 | 1 | 7 |
| $8^i mod\ 10$ | 8 | 4 | 2 | 6 | 8 |
| $9^i mod\ 10$ | 9 | 1 | 9 | 1 | 9 |

(a) Column 5 demonstrates Theorem 6. That is, for every positive integer $x < n$ (every row), column 5 shows $x^{\phi(n)+1} mod\ n = x$.

(b) Column 4 demonstrates Euler's Theorem. From Euler's Theorem, we know that if $x \in Z_n^*$, then $x^{\phi(n)}\ mod\ n = 1$. This explains why column 4 shows a value of 1 for rows {1, 3, 7, 9}, which are highlighted in yellow.

(c) Now why the value in column 4 cannot be 1 for any other row? If column 4 shows a value of 1 for any $x$, that means $x^4 mod\ n = 1$, which implies $x^3\ mod\ n = x^{-1}$. But from Theorem 3, we know $x$ does not have an inverse if $\gcd(x, n) \neq 1$, which means $x \notin Z_n^*$. Therefore, $x^4\ mod\ n \neq 1$ for any $x \notin Z_n^*$.

10. (RSA Public Key Cryptosystem) Let us pick two prime numbers $p = 41, q = 67$. Then,

$$n = pq = 41 * 67 = 2747$$
$$\phi(n) = (p-1)(q-1) = 40 * 66 = 2640$$

We pick a small prime number $s$, with $\gcd\left(s, \phi(n)\right) = 1$.

$$s = 13.$$

Then,

$$t = s^{-1}\ mod\ \phi(n) = 13^{-1}\ mod\ 2640 = 2437.$$

(This inverse is computed by Euclid's algorithm.)

The public keys are $(s, n)$ and the private key $(t, n)$.

(a) Suppose a sender wants to send a number $x = 2169$. Compute the encrypted message $y$. Make sure you perform a mod operation after each multiplication so the intermediate results does not become too large. Use the recursive algorithm Power to compute the exponentiation. (You may use an Excel sheet to generate the data.) Show the computation.

(b) Compute the decrypted message $z$ and verify that $z = x$. Show the computation.

**Solution:**

(a) Encryption ($x = 2169$):

$$y = x^s \; mod \; n = 2169^{13} \; mod \; 2747 = 223$$

Below is the computation of power, using the recursive algorithm. (In the rows where the power $i$ is odd, after the square and mod operation, the result is multiplied by $x$ and followed by a second mod operation.)

| Power $i$ | Square | Mod 2747 | $* \, x$ | Mod 2747 |
|---:|---:|---:|---:|---:|
| 13 | 4,020,025 | 1164 | 2,524,716 | 223 |
| 6 | 6,553,600 | 2005 | 2,005 | 2005 |
| 3 | 4,704,561 | 1697 | 3,680,793 | 2560 |
| 1 | | | | 2169 |

(b) Decryption:

$$z = y^t \; mod \; n = 223^{2437} mod \; 2747 = 2169$$

Below is the computation of the latter exponentiation, using the recursive algorithm.

| Power $i$ | Square | Mod 2747 | $* \, x$ | Mod 2747 |
|---:|---:|---:|---:|---:|
| 2437 | 30,276 | 59 | 13,157 | 2,169 |
| 1218 | 181,476 | 174 | 174 | 174 |
| 609 | 1,656,369 | 2,675 | 596,525 | 426 |
| 304 | 1,819,801 | 1,287 | 1,287 | 1,287 |
| 152 | 1,004,004 | 1,349 | 1,349 | 1,349 |
| 76 | 212,521 | 1,002 | 1,002 | 1,002 |
| 38 | 1,552,516 | 461 | 461 | 461 |
| 19 | 118,336 | 215 | 47,945 | 1,246 |
| 9 | 181,476 | 174 | 38,802 | 344 |
| 4 | 80,089 | 426 | 426 | 426 |
| 2 | 49,729 | 283 | 283 | 283 |
| 1 | | | | 223 |

| CS 506, Online Foundations of CS | **Homework 8 Permutations/Combinations Solution** | Dr. David Nassimi |
|---|---|---|

**Study Module 9 (Counting Methods)**
**Permutations and Combinations**

1. (a) Use the multiplication principle to prove that the total number of $n$-bit binary integers is $2^n$.
   (b) Use induction to prove the same.

**Solution:**

(a) Each bit has 2 possible values. So the total number is

$$2 \times 2 \times 2 \times \cdots \times 2 = 2^n.$$

(b) The base case, $n = 1$, is a 1-bit integer, which obviously has two possible values. Now suppose the number of $n$-bit integers is $f(n) = 2^n$ for some $n \geq 1$. Then, to form $(n+1)$-bit integers, we take each $n$-bit integer and append it with another bit, which may be either 0 or 1. So, the total number of $(n+1)$-bit integers becomes

$$f(n+1) = 2 * f(n) = 2 * 2^n = 2^{n+1}.$$

□

2. (a) Use the addition principle (also called Incusion-Exclusion principle) to derive the number of elements in the union of three sets $A, B, C$, which are not disjoint.
   (b) Repeat for four sets $A, B, C, D$.

**Solution:**

(a)
$$\begin{aligned} |A \cup B \cup C| &= |A| + |B| + |C| \\ &\quad -|A \cap B| - |A \cap C| - |B \cap C| \\ &\quad +|A \cap B \cap C| \end{aligned}$$

(b)
$$\begin{aligned} |A \cup B \cup C \cup D| &= |A| + |B| + |C| + |D| \\ &\quad -|A \cap B| - |A \cap C| - |A \cap D| - |B \cap C| - |B \cap D| - |C \cap D| \\ &\quad +|A \cap B \cap C| + |A \cap B \cap D| + |A \cap C \cap D| + |B \cap C \cap D| \\ &\quad -|A \cap B \cap C \cap D| \end{aligned}$$

□

3. (a) Define $P(n, r)$, and give the formula for it. Compute $P(8, 4)$.
   (b) Define $C(n, r)$, and give the formula for it.
   (c) What is $C(n, r)$ in terms of $P(n, r)$? Explain.
   (d) Compute $C(8, 4)$, $C(8, 2)$, and $C(8, 6)$.

**Solution:**

(a) $P(n, r)$ is the number of ordered subsets of size $r$ out of $n$ distinct elements.

$$P(n, r) = n!/(n-r)!$$

(b) $C(n, r)$ is the number of unordered subsets of size $r$ out of $n$ distinct elements.

$$C(n, r) = \frac{n!}{r!(n-r)!}$$

(c) In $P(n, r)$, each subset has $r!$ permutations and is counted $r!$ times. But in $C(n, r)$, each subset is counted as only one since it is unordered. So,

$$C(n, r) = P(n, r)/r!$$

(d) $C(8, 4) = (8 * 7 * 6 * 5)/(4 * 3 * 2 * 1) = 70$
$C(8, 2) = C(8, 6) = (8 * 7)/((2 * 1) = 28$

□

4. Consider the binomial formula:

$$(a + b)^n = c_0 a^n + c_1 a^{n-1} b + c_2 a^{n-2} b^2 + \cdots + c_n b^n.$$

(a) Compute the coefficients for $(a + b)^4$.

(b) Compute the coefficients for $(a + b)^6$.

**Solution:**

(a) $(a+b)^4 = C(4, 4)a^4 + C(4, 3)a^3 b + C(4, 2)a^2 b^2 + C(4, 1)ab^3 + C(4, 0)b^4$
$= a^4 + 4a^3 b + 6a^2 b^2 + 4ab^3 + b^4$

(b) $(a+b)^6 = C(6, 6)a^6 + C(6, 5)a^5 b + C(6, 4)a^4 b^2 + C(6, 3)a^3 b^3 + C(6, 2)a^2 b^4 + C(6, 1)ab^5 + C(6, 0)b^6$
$= a^6 + 6a^5 b + 15a^4 b^2 + 20a^3 b^3 + 15a^2 b^4 + 6ab^5 + b^6$

□

5. (a) Explain the recursive formula for $C(n, r)$, which is as follows:

$$C(n, r) \quad = \quad \begin{cases} C(n-1, r) + C(n-1, r-1), & 0 < r < n, \\ 1, & r = 0, r = n. \end{cases}$$

(b) Prove by induction that the solution of this recurrence is

$$C(n, r) = \frac{n!}{r!(n-r)!}$$

**Solution:**

(a) To derive the number of ways to choose an $r$ element subset of $\{1, 2, 3, \cdots, n\}$, we note that element 1 is either selected or not selected for the subset.

- If 1 is not in the subset: Then pick all of the $r$ elements out of $\{2, 3, \cdots, n\}$. The number of ways is
$$C(n-1, r).$$
- If 1 is in the subset: Then pick the remaining $r - 1$ out of $n - 1$. Number of ways is
$$C(n-1, r-1).$$

These two sets of choices are disjoint. So, by the addition principle, the total number is the sum of the two. So,

$$C(n, r) = C(n-1, r) + C(n-1, r-1).$$

(b) Proof by induction: First note that $0! = 1$ by definition.
For the induction base, there are two cases $r = 0$ and $r = n$.

- For $r = 0$, $C(n, 0) = 1$ from the recurrence. And from the solution:

$$C(n, 0) = \frac{n!}{0!(n-0)!} = \frac{n!}{n!} = 1.$$

- For $r = n$, it is similar. From the recurrnace $C(n, n) = 1$. And from the solution:

$$C(n, n) = \frac{n!}{n!(n-n)!} = \frac{n!}{n!} = 1.$$

So the solution is correct for both base cases. Next, to prove $C(n, r)$ for any $n$ and $r$, suppose the solution form is correct for all smaller values of $n$ and $r$. In particular,

$$C(n-1, r) = \frac{(n-1)!}{r!(n-1-r)!}$$

$$C(n-1, r-1) = \frac{(n-1)!}{(r-1)!(n-r)!}.$$

Then,

$$\begin{aligned} C(n, r) &= \frac{(n-1)!}{r!(n-1-r)!} + \frac{(n-1)!}{(r-1)!(n-r)!} \\ &= \frac{(n-r)(n-1)! + r(n-1)!}{r!(n-r)!} \\ &= \frac{(n-1)!(n-r+r)}{r!(n-r)!} = \frac{n!}{r!(n-r)!}. \end{aligned}$$

□

6. (a) What is the number of 6-bit binary integers with exactly 3 1's? List them in sorted order.
(b) What is the number of $n$-bit binary integers with exactly $r$ 1's?
(c) Explain the reason for the following equality.

$$\Sigma_{r=0}^{n} C(n, r) = 2^n.$$

**Solution:**

(a) The number is $C(6, 3) = (6 * 5 * 4)/(3 * 2 * 1) = 20$.
List of them:

000111
001011
001101
001110

010011
010101
010110
011001
011010
011100

100011
100101
100110
101001
101010
101100

110001
110010
110100
111000

(b) The number is $C(n, r)$.

(c) The number of $n$ bit integers with exactly $r$ 1's is $C(n, r)$. The sum of this for $r = 0$ to $n$ will include all $n$ bit integers, so the sum is $2^n$.

□

7. (a) A 10-member club wants to choose a president, a vice-president, and a treasurer. Derive the number of ways.

   (b) A 20-member club wants to choose a president, a vice-president, 3 secretaries, and 2 treasurers. Derive the number of ways.

   (c) A 25-member club wants to choose a 5-member committee for fund-raising. Derive the number of ways to choose the committee.

**Solution:**

(a) $P(10, 3) = 10 * 9 * 8 = 720$.

(b) The number is

$$\frac{20!}{1!1!3!2!13!} = \frac{20 * 19 * 18 * 17 * 16 * 15 * 14}{3 * 2 * 2} = 32,558,400.$$

(c) $C(25, 5) = (25 * 24 * 23 * 22 * 21)/(5 * 4 * 3 * 2 * 1) = 53,130$.

□

8. How many distinct strings result from all permutations of the 5 characters in string "ABBCC"? Show the combinatorial derivation. List the distinct strings in alphabetical order.

**Solution:** The number is

$$\frac{5!}{2!2!} = 30.$$

```
AB BCC
AB CBC
AB CCB
------
AC BBC
AC BCB
AC CBB
------
BA BCC
BA CBC
BA CCB
------
BB ACC
BB CAC
BB CCA
------
BC ABC
BC ACB
BC BAC
BC BCA
BC CAB
BC CBA
------
CA BBC
CA BCB
CA CBB
------
CB ABC
CB ACB
CB BAC
CB BCA
CB CAB
CB CBA
------
CC ABB
CC BAB
CC BBA
```

□

9. There are 6 identical golf balls and 3 holes $A$, $B$, and $C$. How many ways are there for placing the balls in the holes?

   Hint: Line up the balls in a row and place two dividers between them. For example, consider the following line up, where $o$ is a ball and $|$ is a divider:

   | Positions: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
   |---|---|---|---|---|---|---|---|---|
   | Line up: | $o$ | $o$ | $o$ | $\mid$ | $o$ | $o$ | $\mid$ | $o$ |

   In this line up, the first 3 balls (to the left of the first divider) go in hole $A$, the next 2 balls go in hole $B$, and the last ball goes in hole $C$. (See Example 6.3.4 and the following theorem in the text.)

   Six balls plus two dividers is 8 places. The number is $C(8,2) = (8 * 7)/2 = 28$.

10. Four friends want to divide a pie of pizza that has 8 identical slices (with no leftover).

    (a) How many ways are there to divide the 8 slices between them, with no restriction?

    (b) How many ways are there if each person gets at least one slice?

**Solution:**

(a) 8 slices plus 3 dividers = 11 places
The number is
$$C(11,3) = (11 * 10 * 9)/(3 * 2 * 1) = 165.$$

(b) Give one slice to each person. Then there are 4 slices remaining. Divide 4 slices among 4 friends. 4 slices plus 3 dividers = 7 places.
$$C(7,3) = 35.$$

□

# Additional Exercises (Not to be handed-in)

11. Show the computations of $C(6,r)$, for $r = 0, 1, 2, 3, 4, 5, 6$. Verify that the total of these values is 64, which is $2^6$. Explain briefly why.

12. Parents of three (spoiled) children bring home nine gifts to divide between the kids for Chanukah/Christmas. The gifts are:

    - A series of three different books on Hunger Games, namely: (1) The Hunger Games, (2) Catching Fire, and (3) Mockingjay.
    - Three distint boxes of chocolates.
    - Three iTunes gift cards, each with the value of \$100.

    Derive the total number of distinct ways to divide the nine gifts beween the three children for each of the following cases:

    (a) Assuming an equitable division of the gifts such that each kid gets one book, one box of chocolate, and one iTunes gift card.

    (b) Assuming no restriction on how many gifts each kid receives. (A bully child may end up with all nine gifts, but that is a valuable lesson for the kids to learn.)

13. Prove by induction that an integer $n$ is divisible by 3 if and only if sum of its decimal digits is divisible by 3. (For example, 47298 is divisible by 3 since $4 + 7 + 2 + 9 + 8 = 30$. What happens to the sum of digits when you add 3 to a number, $47298 + 3 = 47301$? How does each decimal carry effect the sum?)

14. Prove by induction that an integer $n$ is divisible by 9 if and only if sum of its decimal digits is divisible by 9.

| CS 506, Online<br>Foundations of CS | **Homework 7**<br>**Recurrences**<br>**Solution** | Dr. David Nassimi |
|---|---|---|

## Study Module 8 (Classes of Recurrence Equations)
## Divide-and-Conquer Recurrences (Master Theorem) and Linear Recurrences

1. The following class of recurrences often arise in analysis of divide-and-conquer algorithms. (Note: $a, b, c, d$ and $\beta$ are constants; and $n$ is an integer power of $b$, $n = b^k$.)

$$T(n) = \begin{cases} a\ T(n/b) + c\ n^\beta, & n > 1 \\ d, & n = 1 \end{cases}$$

Let $h = \log_b a$. We used *repeated substitution* method to derive the solution:

$$T(n) = \begin{cases} A\ n^h + B\ n^\beta & = \Theta(n^\beta), & h < \beta \\ A\ n^h + B\ n^\beta & = \Theta(n^h), & h > \beta \\ A\ n^h + B\ n^h \log n & = \Theta(n^h \log n), & h = \beta \end{cases} \tag{1}$$

where $A$ and $B$ are some constants for each case. (A slightly different and more general version of this class of recurrences is presented in various textbooks and is commonly known as *Master Theorem.*)

Use the above formula to find the solution form for each of the following recurrences, where $n$ is a power of 2. Express the solution form invloving constants $A$ and $B$. (Don't bother to find the constants.) Then express in $\Theta(\ )$ form.

(a) $T(n) = 2T(n/2) + 1$

$a = 2, b = 2, \beta = 0.\quad h = \log_b a = \log_2 2 = 1.\quad$ Since $h \neq \beta$,

$$T(n) = An^h + Bn^\beta = An + B = \Theta(n).$$

(b) $T(n) = 2T(n/2) + n$

$a = 2, b = 2, \beta = 1.\quad h = \log_b a = \log_2 2 = 1.\quad$ Since $h = \beta$,

$$T(n) = An^h \log n + Bn^\beta = An \log n + Bn = \Theta(n \log n).$$

(c) $T(n) = 2T(n/2) + n^2$

$a = 2, b = 2, \beta = 2.\quad h = \log_b a = \log_2 2 = 1.\quad$ Since $h \neq \beta$,

$$T(n) = An^h + Bn^\beta = An + Bn^2 = \Theta(n^2).$$

(d) $T(n) = T(n/2) + 1$

$a = 1, b = 2, \beta = 0.\quad h = \log_2 1 = 0.\quad$ Since $h = \beta$,

$$T(n) = An^h \log n + Bn^\beta = A \log n + B = \Theta(\log n).$$

2. Consider the following recurrence, where $n$ is a power of 2, $T(1) = 0$, and

$$T(n) = 4T(n/2) + n, \ n > 1.$$

(a) Find the exact solution by repeated substitution.

$$\boxed{\begin{aligned} T(n) &= n + 4T(n/2) \\ &= n + 4(n/2 + 4T(n/4)) \\ &= n + 2n + 4^2T(n/2^2) \\ &= n + 2n + 2^2n + 4^3T(n/2^3) \\ &= n + 2n + 2^2n + \cdots + 2^{k-1}n + 4^kT(n/2^k), \quad \text{where } n = 2^k \\ &= n(1 + 2 + 4 + \cdots + 2^{k-1}) + 4^kT(1), \quad \text{where } T(1) = 0 \\ &= n(2^k - 1) = n(n-1) = n^2 - n. \end{aligned}}$$

(b) Find the exact solution by using the above (Master Theorem) formula to obtain the solution form, and then find the constants $A$ and $B$.

We have $a = 4, b = 2, \beta = 1$. Thus, $h = \log_2 4 = 2$. Since $h \neq \beta$,

$$T(n) = An^h + Bn^\beta = An^2 + Bn.$$

The constants $A$ and $B$ may be found by using two values of $n = 1, 2$.

$$\begin{aligned} T(1) &= 0 = A + B, \\ T(2) &= 2 + 4T(1) = 2 = 4A + 2B. \end{aligned}$$

Thus, $A = 1, B = -1$. That is, $T(n) = n^2 - n$.

3. Consider the following recurrence equation, where $n$ is a power of 2.

$$T(n) = \begin{cases} 4T(n/2) + n + 6, & n \geq 2 \\ 0, & n = 1 \end{cases}$$

Prove by induction that the solution is

$$T(n) = An^2 + Bn + C$$

and find the constants $A, B, C$.

Proof: For the base, $n = 1$,

$$T(1) = 0 = A + B + C$$

To prove the solution for any $n \geq 2$, suppose the solution is correct for $n/2$. That is,

$$T(n/2) = A(n/2)^2 + B(n/2) + C \quad \text{hypothesis}$$

Then,

$$\begin{aligned} T(n) &= 4T(n/2) + n + 6 && \text{from the recurrence} \\ &= 4[An^2/4 + Bn/2 + C] + n + 6 && \text{from the hypothsis} \\ &= An^2 + (2B+1)n + (4C+6) \\ &= An^2 + Bn + C && \text{To reach the conclusion} \end{aligned}$$

To satisfy the latter equality (to reach the conclusion), we need to equate term-by-term.

$$2B + 1 = B$$

$$4C + 6 = C$$

So we have three equations for $A, B, C$:

$$A + B + C = 0; \quad 2B + 1 = B; \quad 4C + 6 = C$$

We solve for the constants and get

$$B = -1, C = -2, A = 3$$

Therefore we have proved that

$$T(n) = 3n^2 - n - 2.$$

4. Find the solution of each of the following linear recurrences.

(a)

$$F_n = \begin{cases} 3, & n = 0 \\ 29, & n = 1 \\ 10F_{n-1} - 21F_{n-2}, & n \geq 2 \end{cases}$$

$$\begin{aligned}
&F_n - 10F_{n-1} + 21F_{n-2} = 0\\
&F_n = r^n\\
&r^n - 10r^{n-1} + 21r^{n-2} = 0\\
&r^{n-2}(r^2 - 10r + 21) = 0\\
&(r-7)(r-3) = 0\\
&r_1 = 7; \quad r_2 = 3\\
&F_n = A7^n + B3^n
\end{aligned}$$

Use base cases (initial values) to find $A, B$.

$$\begin{aligned}
&F_0 = 3 = A + B\\
&F_1 = 29 = 7A + 3B
\end{aligned}$$

We find $A = 5, B = -2$. Therefore,

$$F_n = 5 * 7^n - 2 * 3^n.$$

(b)

$$F_n = \begin{cases} 3, & n = 0\\ 50, & n = 1\\ 10F_{n-1} - 25F_{n-2}, & n \geq 2 \end{cases}$$

$$\begin{aligned}
&F_n - 10F_{n-1} + 25F_{n-2} = 0\\
&F_n = r^n\\
&r^n - 10r^{n-1} + 25r^{n-2} = 0\\
&r^2 - 10r + 25 = 0\\
&(r-5)^2 = 0\\
&r_1 = r_2 = 5
\end{aligned}$$

Since we have reapted roots, the solution is:

$$F_n = A5^n + Bn5^n$$

Use base cases (initial values) to find $A, B$.

$$\begin{aligned}
&F_0 = 3 = A\\
&F_1 = 50 = 5A + 5B
\end{aligned}$$

So, $A = 3, B = 7$. Therefore,

$$F_n = 3 * 5^n + 7 * n5^n.$$

5. A linear recurrence of order 3 has the following initial values.

$$F_0 = 4; \quad F_1 = 19; \quad F_2 = 69$$

The recurrence equation has the following characteristic equation in factored form.

$$(r-2)^2(r-5) = 0.$$

(a) Find the complete solution.

Roots are $r_1 = r_2 = 2, \quad r_3 = 5$. Therefore, the complete solution is

$$F_n = A2^n + Bn2^n + C5^n$$

Use the base cases to find the constants $A, B, C$.

$$\begin{aligned}&F_0 = 4 = A + C\\&F_1 = 19 = 2A + 2B + 5C\\&F_2 = 69 = 4A + 8B + 25C\end{aligned}$$

So, $A = 3, B = 4, C = 1$. Therefore,

$$F_n = 3 * 2^n + 4 * n2^n + 5^n.$$

(b) Work backward from the characteristic equation to find the recurrence equation.

$$\begin{aligned}&(r-2)^2(r-5) = 0\\&(r^2 - 4r + 4)(r-5) = 0\\&r^3 - 9r^2 + 24r - 20 = 0\\&r^n - 9r^{n-1} + 24r^{n-2} - 20r^n n - 3 = 0\\&F_n - 9F_{n-1} + 24F_{n-2} - 20F_{n-3} = 0\\&F_n = 9F_{n-1} - 24F_{n-2} + 20F_{n-3}.\end{aligned}$$

## Additional Exercises (Not to be handed-in)

6. Consider the following recurrence equation.

$$T(n) = \begin{cases} 2T(n/2) + \log n, & n \geq 2 \\ 0, & n = 1 \end{cases}$$

(a) Try to prove by induction that the solution is

$$T(n) = An + B\log n$$

You will see the induction will fail because you get a constant term on one side but not the other side of an equality. That suggests to refine our guess by adding a constant.

(b) Now, prove by induction that the solution is as follows.

$$T(n) = An + B\log n + C.$$

The above procedure illustrates how we may use master theorem to guide us in guessing the solution when the recurrence equation is not exactly in the form required by our version of master theorem. In this case, the term at the end, which is called "forcing function", is $\log n$ and not the form $n^\beta$.

How did we come up with the above guess in the first place? Observe that

$$h = \log_b a = \log_2 2 = 1$$

So we first guessed the solution as

$$T(n) = An + B * \text{ (forcing function) } = An + B \log n.$$

But the induction proof failed because we were missing a constant to make the final equality. So we refined our guess by adding the constant $C$.

7. **Algebra Facts:**

   (a) Prove that for every integer $n \geq 2$,

   $$\lfloor \log \lfloor n/2 \rfloor \rfloor = \lfloor \log n \rfloor - 1$$

   Hint: Let $2^k \leq n < 2^{k+1}$. Then:

   $$\begin{gathered} k \leq \log n < k+1, \\ \lfloor \log n \rfloor = k. \end{gathered}$$

   And,

   $$\begin{gathered} 2^{k-1} \leq n/2 < 2^k \\ 2^{k-1} \leq \lfloor n/2 \rfloor < 2^k, \\ k-1 \leq \log \lfloor n/2 \rfloor < k, \\ \lfloor \log \lfloor n/2 \rfloor \rfloor = k-1. \end{gathered}$$

   (b) Prove that for every integer $n \geq 2$,

   $$\lceil \log \lceil n/2 \rceil \rceil = \lceil \log n \rceil - 1.$$

   Hint: Let $2^k < n \leq 2^{k+1}$. Prove in a similar way.

8. Consider the following recursive version of binary-search algorithm.
   The original call to this is BS($A, 0, n-1, X$).

```
int BS (dtype $A$[ ], int $i$, int $j$, dtype $X$)
{ //Search $A[i..j]$ for key $X$
1   if ($i > j$) return -1; //Not found if $n = 0$
2   $m = \lfloor (i+j)/2 \rfloor$;
3   if ($X == A[m]$) return $m$;
4   else if ($X < A[m]$)
5       return BS($A, i, m-1, X$);
6   else
7       return BS($A, m+1, j, X$);
}
```

   Let $f(n)$ be the worst-case number of key comparisons to search an array of $n$ elements. Let us count the key comparisons in lines 3 and 4 as one comparison. (We explained the justification in class.) Then,

$$f(n) = \begin{cases} 1, & n = 1 \\ f(\lfloor n/2 \rfloor) + 1, & n > 1. \end{cases}$$

(a) Use the recurrence equation to compute and tabulate $f(n)$ for $n = 1$ to 15.

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $f(n)$ | 1 | 2 | 2 | 3 | 3 | 3 | 3 |

(b) Prove by induction that the solution is

$$f(n) = \lfloor \log_2 n \rfloor + 1.$$

Hint: Use the algebra fact proved above.

**Proof:** For $n = 1$, $f(1) = 1$ from the recurrence. And, the claimed solution is: $f(1) = \log 1 + 1 = 1$. Thus, the solution is correct for $n = 1$.
Now, we will prove the solution is correct for any $n \geq 2$, supposing that $f(m) = \lfloor \log_2 m \rfloor + 1, \ \forall m < n$.

$$\begin{aligned} f(n) &= f(\lfloor n/2 \rfloor) + 1 \\ &= \lfloor \log \lfloor n/2 \rfloor \rfloor + 1 + 1 \quad \text{by hypothesis} \\ &= (k - 1) + 1 + 1 \\ &= k + 1 \\ &= \lfloor \log n \rfloor + 1. \end{aligned}$$

9. Given a sorted array $A$ of $n$ elements, and a search key $X$. Modify the binay-search algorithm to find the left-most occurence of $X$ in the array, in case there are several elements equal to $X$. For example, for the following array, a search key $X = 5$ should return index 3.

| $i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $A[i]$ | 1 | 2 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 10 | 20 |

The algorithm must be asymptotically efficient, and have worst-case time $O(\log n)$, even if the entire array consists of values equal to $X$. (An algorithm that performs partly sequential search and thus has worst-case time $O(n)$ is completely unacceptable.)

(a) Write a recursive version of the modified binary search algorithm.

(b) Illustrate your algorithm for the above example array.

(c) Let $f(n)$ be the worst-case number of key comparisons to search an array of $n$ elements. Write a recurrence for $f(n)$.

(d) Use the recurrence to compute and tabulate $f(n)$ for $n = 1$ to 16.

(e) Guess the exact solution for $f(n)$ and prove the correctness by induction.

**Solution:** There are several ways of writing this algorithm. Below is probably the best algorithm.

```
int BS (dtype $A[\,]$, int $i$, int $j$, dtype $X$) {
//Search $A[i..j]$ for key $X$
if ($i == j$) //case $n = 1$ terminates recursion
   { if ($X == A[i]$) return $i$;
   else return -1; };
$m = \lfloor (i+j)/2 \rfloor$
if ($X \le A[m]$)
   return BS($A, i, m, X$)
else
   return BS($A, m+1, j, X$)
}
```

Recurrence:

$$f(n) = \begin{cases} 1, & n = 1 \\ f(\lceil n/2 \rceil) + 1, & n > 1. \end{cases}$$

Computed Values:

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $f(n)$ | 1 | 2 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |

From the computed values in the above table, it is easy to see the solution is:

$$f(n) = \lceil \log n \rceil + 1.$$

Proof by induction: Let

$$2^{k-1} < n \le 2^k.$$

Then,

$$2^{k-2} < \lceil n/2 \rceil \le 2^{k-1}.$$

Induction base, $n = 1$, $f(1) = 1 = \lceil \log 1 \rceil + 1$.
Induction step:

$$\begin{aligned} f(n) &= f(\lceil n/2 \rceil) + 1 \quad \text{(from the recurrence)} \\ &= \lceil \log \lceil n/2 \rceil \rceil + 1 + 1 \quad \text{(from the hypothesis)} \\ &= (k-1) + 2 \\ &= k + 1 = \lceil \log n \rceil + 1. \end{aligned}$$

□

10. Given two sorted arrays $A$ and $B$ of size $s$ and $t$, respectively. We want to merge them into array $C$ of size $n = s + t$.

    (a) Write a non-recursive program to do the merge.
    merge (dtype $A[\,]$, int $s$, dtype $B[\,]$, int $t$, dtype $C[\,]$)

```
{
int $i, j, k$;
$i = 0; j = 0; k = 0;$
while ($i \le s-1$ and $j \le t-1$)
   { if ($A[i] \le B[j]$)
      $C[k++] = A[i++]$;
   else
```

```
        $C[k++] = B[j++]$;
    }
    while ($i \leq s-1$)
        $C[k++] = A[i++]$;
    while ($j \leq t-1$)
        $C[k++] = B[j++]$;
    }
```

(b) Write a recursive version of the program to do the merge.

Hint: You can formulate the recursion in such a way that on a recursive call, the arrays to be merged always start at index 0. That would eliminate the need for additional parameters. To accomplish this, the algorithm starts by comparing the largest element of $A$ with the largest element of $B$.

```
merge (dtype $A[\,]$, int $s$, dtype $B[\,]$, int $t$, dtype $C[\,]$)
{
if ($s == 0$ and $t == 0$) return;
if ($s > 0$ and $t > 0$) {
    if ($A[s-1] > B[t-1]$)
        { $C[s+t-1] = A[s-1]$;   $s--$ };
    else
        { $C[s+t-1] = B[t-1]$;   $t--$ };
    merge($A, s, B, t, C$);
    };
else {
    while ($s > 0$)
        { $C[s-1] = A[s-1]$;     $s--$ };
    while ($t > 0$)
        { $C[t-1] = B[t-1]$;     $t--$ };
    }
}
```

(c) Let $f(n)$ be the worst-case number of key comparisons to merge two sorted arrays with combined length of $n$. Write a recurrence for $f(n)$. Then find the solution by repeated substitution.

$$f(n) = \begin{cases} 0, & n = 1 \\ 1 + f(n-1), & n > 1 \end{cases}$$

Solution:

$$f(n) = 1 + f(n-1) = 1 + 1 + f(n-2) = 3 + f(n-3) = \cdots = n - 1 + f(1) = n - 1.$$

| CS 506, Online Foundations of CS | **Homework 6 Algorithms Solution** | Dr. David Nassimi |
|---|---|---|

**Study Module 6 (Analysis of Algorithms) and Module 7 (Recursive Algorithms)**

1. Prove the following polynomial is $\Theta(n^3)$. That is, prove $T(n)$ is both $O(n^3)$ and $\Omega(n^3)$.

$$T(n) = 2n^3 - 10n^2 + 100n - 50$$

   (a) Prove $T(n)$ is $O(n^3)$: By definition, you must find positive constants $C_1$ and $n_0$ such that

$$T(n) \le C_1 n^3, \quad \forall n \ge n_0.$$

   (b) Prove $T(n)$ is $\Omega(n^3)$: By definition, you must find positive constants $C_2$ and $n_0$ such that

$$T(n) \ge C_2 n^3, \quad \forall n \ge n_0.$$

Note: Since the highest term in $T(n)$ is $2n^3$, it is possible to pick $n_0$ large enough so that $C_1$ and $C_2$ are close to the coefficient 2. (The definitions of $O()$ and $\Omega()$ are not concerned with this issue.) For this problem, you are required to pick $n_0$ so that $C_1$ and $C_2$ fall within 10% of the coefficient 2. That is,

$$C_2 n^3 \le T(n) \le C_1 n^3, \quad \forall n \ge n_0$$

where $C_2 \ge 1.8$ and $C_1 \le 2.2$.

**Solution:**

- Prove $O(n^3)$.

$$\begin{array}{rcll} T(n) & = & 2n^3 - 10n^2 + 100n - 50 & \\ & \le & 2n^3 + 100n, & n \ge 0 \quad \text{(drop negative terms)} \\ & \le & n^3(2 + 100/n^2), & \\ & \le & n^3(2 + 100/100^2), & n \ge 100 \quad \text{(The choice of } n \ge 100 \text{ is arbitrary.)} \\ & \le & (2.01)n^3, & n \ge 100 \end{array}$$

- Prove $\Omega(n^3)$.

$$\begin{array}{rcll} T(n) & = & 2n^3 - 10n^2 + 100n - 50 & \\ & \ge & 2n^3 - 10n^2 - 50, & n \ge 0 \quad \text{(drop positive terms)} \\ & \ge & n^3(2 - 10/n - 50/n^3), & \\ & \ge & n^3(2 - 10/100 - 50/100^3), & n \ge 100 \quad \text{(The choice of } n \ge 100 \text{ is arbitrary.)} \\ & \ge & n^3(2 - .1 - .00005) & \\ & \ge & (1.89995)n^3 & n \ge 100 \end{array}$$

$\square$

2. (a) Compute and tabulate the following functions for $n = 1, 2, 4, 8, 16, 32, 64$. The purpose of this exercise is to get a feeling for these growth rates and how they compare with each other. (All logarithms are in base 2, unless stated otherwise.)

$$\log n, \; n, \; n \log n, \; n^2, \; n^3, \; 2^n.$$

| $n$ | $\log n$ | $n \log n$ | $n^2$ | $n^3$ | $2^n$ |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 2 |
| 2 | 1 | 2 | 4 | 8 | 4 |
| 4 | 2 | 8 | 16 | 64 | 16 |
| 8 | 3 | 24 | 64 | 512 | 256 |
| 16 | 4 | 64 | 256 | 4,096 | 65,536 |
| 32 | 5 | 160 | 1,024 | 32,768 | 4,294,967,296 |
| 64 | 6 | 384 | 4,096 | 262,144 | $\approx 18.45 \times 10^{18}$ |

(b) Order the following complexity functions (growth rates) from the smallest to the largest. That is, order the functions asymptotically. Note that $\log^2 n$ means $(\log n)^2$.

$$n^2 \log n,\ 5,\ n \log^2 n,\ 2^n,\ n^2,\ n,\ \sqrt{n},\ \log n,\ \frac{n}{\log n}$$

**Solution:** The growth rates are ordered as follows.

$$5,\ \log n,\ \sqrt{n},\ \frac{n}{\log n},\ n,\ n \log^2 n,\ n^2,\ n^2 \log n,\ 2^n.$$

For example, consider the comparison between $n \log^2 n$ and $n^2$. We may apply the ratio test (and La'Hospital rule of calculus):

$$\begin{aligned}\lim_{n\to\infty} \frac{n \log^2 n}{n^2} &= \lim_{n\to\infty} \frac{\log^2 n}{n} = \lim_{n\to\infty} \frac{2(\log n)(1/n)}{1} \\ &= \lim_{n\to\infty} \frac{2 \log n}{n} = \lim_{n\to\infty} \frac{2(1/n)}{1} \\ &= \lim_{n\to\infty} \frac{2}{n} = 0.\end{aligned}$$

The ratio test concludes that $n \log^2 n$ is asymptotically smaller than $n^2$. □

3. Find the exact number of times (in terms of $n$) the innermost statement ($X = X + 1$) is executed in the following code. That is, find the final value of $X$. Then express the total running time in terms of $O(\ )$, $\Omega(\ )$, or $\Theta(\ )$ as appropriate.

```
$X = 0$;
for $k = 1$ to $n$
    for $j = 1$ to $n - k$
      $X = X + 1$;
```

**Solution:** The total number of times the innermost statement is executed is:

$$f(n) = \sum_{k=1}^{n} \sum_{j=1}^{n-k} (1) = \sum_{k=1}^{n} (n-k) = \sum_{k=1}^{n-1} k = n(n-1)/2 = \frac{n^2 - n}{2}.$$

The total running time is $\Theta(f(n)) = \Theta(n^2)$. □

4. The following program computes and returns $(\log_2 n)$, assuming the input $n$ is an integer power of 2. That is, $n = 2^j$ for some integer $j \geq 0$.

```
int LOG (int $n$){
int $m, k$;
$m = n$;
$k = 0$;
while ($m > 1$) {
   $m = m/2$;
   $k = k + 1$; }
ruturn ($k$)
}
```

(a) First, trace the execution of this program for a specific input value, $n = 16$. Tabulate the values of $m$ and $k$ at the beginning, just before the first execution of the while loop, and after each execution of the while loop.

(b) Prove by induction that at the end of each execution of the while loop, the following relation holds between variables $m$ and $k$. (This relation between the variables is called the *loop invariant.*)

$$m = n/2^k.$$

(c) Then conclude that at the end, after the last iteration of the while loop, the program returns $k = \log_2 n$.

**Solution:**

(a)

| $m$ | $k$ |
|---|---|
| 16 | 0 |
| 8 | 1 |
| 4 | 2 |
| 2 | 3 |
| 1 | 4 |

(b) Intitially, $(m = n, k = 0)$ so $m = n/2^k$.
Now suppose at the start of some iteration, $m = n/2^k$ holds true. At the end of that iteration, both sides get divided by two, so the relation still holds. More formally, let $(m', k')$ be the updated values of $(m, k)$ at the end of that iteration. Then, $m' = m/2$ and $k' = k + 1$. So,

$$m' = m/2 = n/2^{k+1} = n/2^{k'}$$

so the relation still will hold.

(c) After the last iteration, $m = 1 = n/2^k$. Therefore, $n = 2^k$, which means $k = \log_2 n$.

□

5. The following pseudocode computes the sum of an array of $n$ integers.

```
int sum (int $A[\,]$, int $n$) {
T = A[0];
for $i = 1$ to $n - 1$
   $T = T + A[i]$;
return $T$;
}
```

(a) Write a recursive version of this code.

(b) Let $f(n)$ be the number of additions performed by this computation. Write a recurrence equation for $f(n)$. (Note that the number of addition steps should be exactly the same for both the non-recursive and recursive versions. In fact, they both should make exactly the same sequence of addition steps.)

(c) Prove by induction that the solution of the recurrence is $f(n) = n - 1$.

**Solution:**

(a)
```
int sum (int $A[\,]$, int $n$) {
if ($n == 1$) return $A[0]$;
$T$ = sum $(A, n-1)$;     //compute sum of $n-1$ elements recursively
return $(T + A[n-1])$;     //add last element
}
```

(b)

$$f(n) = \begin{cases} 0, & n = 1 \\ f(n-1) + 1, & n > 1 \end{cases}$$

(c) For the base, $n = 1$, $f(1) = 0$ and $f(n) = n - 1 = 1 - 1 = 0$, so base is correct.
To prove the solution for any $n \geq 2$, suppose that $f(n-1) = n - 2$. Then,

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

□

6. The following pseudocode finds the maximum element in an array of size $n$.

```
int MAX (int $A[\,]$, int $n$) {
M = A[0];
for $i = 1$ to $n-1$
   if ($A[i] > M$)
      $M = A[i]$     // Update the max
return $M$;
}
```

(a) Write a recursive version of this program.

(b) Let $f(n)$ be the number of key comparisons performed by this algorithm. Write a recurrence equation for $f(n)$.

(c) Prove by induction that the solution of the recurrence is $f(n) = n - 1$.

**Solution:**

(a)
```
int MAX (int $A[\,]$, int $n$) {
if ($n == 1$) return $A[0]$;
M = MAX$(A, n-1)$;
if ($A[n-1] > M$)
   $M = A[n-1]$
return $M$;
}
```

(b)

$$f(n) = \begin{cases} 0, & n = 1 \\ f(n-1) + 1, & n > 1 \end{cases}$$

(c) For the base, $n = 1$, $f(1) = 0$ and $f(n) = n - 1 = 1 - 1 = 0$, so base is correct.
To prove the solution for any $n \geq 2$, suppose that $f(n-1) = n - 2$. Then,

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

□

7. Consider the following pseudocode for insertion-sort algorithm. The algorithm sorts an arbitrary array $A[0..n-1]$ of $n$ elements.

```
void ISORT (dtype $A$[ ], int $n$)
{ int $i, j$;
for $i = 1$ to $n - 1$
    { // Insert $A[i]$ into the sorted part $A[0..i-1]$
    $j = i$;
    while ($j > 0$ and $A[j] < A[j-1]$) {
        SWAP ($A[j], A[j-1]$);
        $j = j - 1$ }
    }
}
```

(a) Illustrate the algorithm on the following array by showing each comparison/swap operation. What is the total number of comparisons made for this worst-case data?

$$A = (5, 4, 3, 2, 1)$$

(b) Write a recursive version of this algorithm.

(c) Let $f(n)$ be the worst-case number of key comparisons made by this algorithm to sort $n$ elements. Write a recurrence equation for $f(n)$. (Note that the sequence of comparisons are exactly the same for both non-recursive and recursive versions. But, you may find it more convenienet to write the recurrence for the recursive version.)

(d) Find the solution for $f(n)$ by repeated substitution.

**Solution:**

(a)

| | | | | | |
|---|---|---|---|---|---|
| (5 | 4) | 3 | 2 | 1 | Insert 4 |
| 4 | (5 | 3) | 2 | 1 | Insert 3 |
| (4 | 3) | 5 | 2 | 1 | |
| 3 | 4 | (5 | 2) | 1 | Insert 2 |
| 3 | (4 | 2) | 5 | 1 | |
| (3 | 2) | 4 | 5 | 1 | |
| 2 | 3 | 4 | (5 | 1) | Insert 1 |
| 2 | 3 | (4 | 1) | 5 | |
| 2 | (3 | 1) | 4 | 5 | |
| (2 | 1) | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |

Total number of compare/swaps for this worst-case ($n = 5$) is 10.

(b) Recursive ISORT:

```
void ISORT (dtype $A[\,]$, int $n$)
if ($n == 1$) return;
ISORT ($A, n-1$);
Sort $n-1$ elements recursively
// Now insert the last element, $A[n-1]$, into the sorted part $A[0..n-2]$
$j = n-1$;
while ($j > 0$ and $A[j] < A[j-1]$) {
   SWAP ($A[j], A[j-1]$);
   $j = j-1$ }
}
```

The initial call is ISORT($A, n$).

(c) $f(n) = \begin{cases} 0, & n = 1 \\ f(n-1) + n - 1 & n > 1 \end{cases}$

(d) Repeated Substitution method:

$$\begin{aligned} f(n) &= n - 1 + f(n-1) \\ &= (n-1) + (n-2) + f(n-2) \\ &= (n-1) + (n-2) + (n-3) + f(n-3) \\ &\vdots \\ &= (n-1) + (n-2) + \cdots + 1 + f(1) \\ &= (n-1) + (n-2) + \cdots + 1 \\ &= (n-1)(n)/2 \\ &= \frac{n^2 - n}{2} \end{aligned}$$

□

8. Consider the bubble-sort algorithm described below.

```
void bubble (dtype $A[\,]$, int $n$)
{ int $i, j$;
for ($i = n-1$;  $i > 0$;  $i--$)                    //Bubble max of $A[0..i]$ down to $A[i]$.
   for ($j = 0$;  $j < i$;  $j++$)
      if ($A[j] > A[j+1]$)    SWAP($A[j], A[j+1]$);
}
```

(a) Analyze the time complexity, $T(n)$, of the bubble-sort algorithm.

**Solution:**

- **Approach 1:** We can first find the total number of key-comparisons.

$$f(n) = \sum_{i=1}^{n-1} \sum_{j=0}^{i-1} (1) = \sum_{i=1}^{n-1} i = \frac{(n-1)n}{2} = \frac{(n^2 - n)}{2} = O(n^2).$$

(This is the number of key-comparisons in every case, not just the worst-case.) Thus, the total running time is

$$T(n) \le Cf(n).$$

Therefore, $T(n) = O(n^2)$.

- **Approach 2:** If we are not interested in finding the exact number of key comparisons, but only want the order of the time, a slightly simpler analysis is as follows. First, observe that the inner for-loop running time is $O(i)$. So, the total time is:

$$T(n) = \sum_{i=1}^{n-1} O(i) = O(\sum_{i=1}^{n-1} i) = O(\frac{(n-1)n}{2}) = O(n^2).$$

- **Approach 3:** A yet simpler analysis is possible that avoids computing the exact summation.
  i. The outer loop is executed $O(n)$ times.
  ii. The inner for-loop has time complexity $O(i)$, which is certainly $O(n)$.

  Thus, by the product rule, $T(n) = O(n) * O(n) = O(n^2)$.
  Note: The type of simplification that was done in the latter step (ii) may sometimes result in over-estimation, though it worked fine in this example.

□

(b) Rewrite the algorithm using recursion.

```
void bubble (dtype $A[\,]$, int $n$) {
if ($n == 1$) return;
for $j = 0$ to $n-2$    // Bubble max to the bottom
   if ($A[j] > A[j+1]$)    SWAP ($A[j], A[j+1]$);
bubble ($A, n-1$);        // Recursive call to sort $n-1$ elements.
}
```

(c) Let $f(n)$ be the worst-case number of key-comparisons used by this algorithm to sort $n$ elements. Write a recurrence for $f(n)$. Solve the recurrence by repeated substitution (i.e, iteration method).

$$f(n) = \begin{cases} 0, & n = 1 \\ n-1+f(n-1), & n > 1 \end{cases}$$

$$\begin{aligned} f(n) &= n-1+f(n-1) \\ &= (n-1)+(n-2)+f(n-2) \\ &= (n-1)+(n-2)+(n-3)+f(n-3) \\ &\vdots \\ &= (n-1)+(n-2)+\cdots+1+f(1) \\ &= (n-1)+(n-2)+\cdots+1 \\ &= (n-1)(n)/2 \\ &= \frac{n^2-n}{2} \end{aligned}$$

9. The following algorithm uses a **divide-and-conquer** technique to find the maximum element in an array of size $n$. The initial call to this recursive function is max(arrayname, 0, n).

```
dtype Findmax(dtype $A[\,]$, int $i$, int $n$)
{    //$i$ is the starting index, and $n$ is the number of elements.
dtype $Max1, Max2$;
if ($n == 1$) return $A[i]$;
$Max1 =$ Findmax ($A, i, \lfloor n/2 \rfloor$);                //Find max of the first half
$Max2 =$ Findmax ($A, i + \lfloor n/2 \rfloor, \lceil n/2 \rceil$);        //Find max of the second half
if ($Max1 \geq Max2$)    return $Max1$;
     else return $Max2$;
}
```

Let $f(n)$ be the worst-case number of *key comparisons* for finding the max of $n$ elements.

(a) Assuming $n$ is a power of 2, write a recurrence relation for $f(n)$. Find the solution by each of the following methods.

i. Apply the repeated substitution method.

ii. Apply induction to prove that $f(n) = An + B$ and find the constants $A$ and $B$.

**Solution:**

$$f(n) = \begin{cases} 0, & n = 1 \\ 2f(n/2) + 1, & n > 1 \end{cases}$$

i. Repeated substitution:

$$\begin{aligned}
f(n) &= 1 + 2f(n/2) \\
&= 1 + 2[1 + 2f(n/4)] \\
&= 1 + 2 + 4f(n/4) \\
&= 1 + 2 + 4 + 8f(n/8) \\
&= 1 + 2 + 4 + \cdots + 2^{k-1} + f(n/2^k), \quad \text{where } n = 2^k \\
&= 1 + 2 + 4 + \cdots + 2^{k-1} + f(1) \\
&= 1 + 2 + 4 + \cdots + 2^{k-1} \qquad \text{Use geometric series formula} \\
&= 2^k - 1 \\
&= n - 1.
\end{aligned}$$

ii. Prove by induction that the solution is $f(n) = An + B$.
For the base, $n = 1$, $f(1) = 1 = A + B$.
For any $n > 1$, supposing that $f(n/2) = A * n/2 + B$, we will prove that $f(n) = An + B$.

$$\begin{aligned}
f(n) &= 2f(n/2) + 1 && \text{(from the recurrence)} \\
&= 2(An/2 + B) + 1 && \text{(from the hypothesis)} \\
&= An + 2B + 1 \\
&= An + B && \text{(needed for the induction step)}
\end{aligned}$$

The latter equality holds if: $2B + 1 = B$. So we have established two relations:

$$\begin{aligned}
A + B &= 1 && \text{(to satisfy the base)} \\
2B + 1 &= B && \text{(to satisfy the induction step)}
\end{aligned}$$

which determine the two unknowns: $A = 1,\ B = -1$. Thus, $f(n) = n - 1$.

$\square$

(b) Now consider the general case where $n$ is any integer. Write a recurrence for $f(n)$. Use induction to prove that the solution is $f(n) = n - 1$.

**Solution:**

$$f(n) = \begin{cases} 0, & n = 1 \\ f(\lfloor n/2 \rfloor) + f(\lceil n/2 \rceil) + 1, & n > 1 \end{cases}$$

**Induction Proof (strong induction):**
For the base, $n = 1$, $f(1) = 0 = n - 1$, thus the base is correct.
For any $n > 1$, supposing that (strong hypothesis):

$$f(m) = m - 1, \quad \forall m < n$$

we will prove that $f(n) = n - 1$.

$$\begin{aligned} f(n) &= f(\lfloor n/2 \rfloor) + f(\lceil n/2 \rceil) + 1 && \text{from the recurrence} \\ &= (\lfloor n/2 \rfloor - 1) + (\lceil n/2 \rceil - 1) + 1 && \text{from the hypothesis} \\ &= \lfloor n/2 \rfloor + \lceil n/2 \rceil - 1 \\ &= n - 1. \end{aligned}$$

□

10. The following divide-and-conquer algorithm is designed to return TRUE if and only if all elements of the array have equal values. For simplicity, suppose the array size is $n = 2^k$ for some integer $k$. Input $S$ is the starting index, and $n$ is the number of elements starting at $S$. The initial call is SAME$(A, 0, n)$.

```
Boolean SAME (int $A$[ ], int S, int n) {
Boolean $T1, T2, T3$;
if ($n == 1$) return TRUE;
$T1$ = SAME ($A, S, n/2$);
$T2$ = SAME ($A, S + n/2, n/2$);
$T3$ = ($A[S] == A[S + n/2]$);
return ($T1 \wedge T2 \wedge T3$);
}
```

(a) Explain how this program works.

(b) Prove by induction that the algorithm returns TRUE if and only if all elements of the array have equal values.

(c) Let $f(n)$ be the number of key comparisons in this algorithm for an array of size $n$. Write a recurrence for $f(n)$.

(d) Find the solution by repeated substitution

**Solution:**

(a) If all elements in the first half are equal ($T1$), and all elements in the second half are equal ($T2$), and an element of the first half is equal to an element of the second half ($T3$), then all elements are equal and the algorithm returns true.

(b) For $n = 1$, there is a single element, which equals itself, so the algorithm correctly returns a true value.
For the induction step, we will prove the algorithm works for any $n \geq 2$, supposing it works for $n/2$. By the hypothesis, the first recursive call returns true in $T1$ iff all elements of the first half are equal. Similarly, the second recursive call returns true in $T2$ iff all elements in the second half are equal. And $T3$ will be true iff an elements of the first half equals an element of the second half. So the algorithm will correctly return true iff all elements are equal.

(c)

$$f(n) = \begin{cases} 0, & n = 1 \\ 2f(n/2) + 1, & n > 1 \end{cases}$$

(d) Repeated substitution:

$$\begin{aligned}
f(n) &= 1 + 2f(n/2) \\
&= 1 + 2[1 + 2f(n/4)] \\
&= 1 + 2 + 4f(n/4) \\
&= 1 + 2 + 4 + 8f(n/8) \\
&= 1 + 2 + 4 + \cdots + 2^{k-1} + f(n/2^k), \quad \text{where } n = 2^k \\
&= 1 + 2 + 4 + \cdots + 2^{k-1} + f(1) \\
&= 1 + 2 + 4 + \cdots + 2^{k-1} \qquad \text{Use geometric series formula} \\
&= 2^k - 1 \\
&= n - 1.
\end{aligned}$$

□

# Additional Exercises (Not to be handed-in)

11. One of the earlier problems above presented the program to compute $\log_2 n$ when input is $n = 2^j$ for some integer $j$.

   (a) Generalize this algorithm to compute $\lfloor \log_2 n \rfloor$ where input $n$ is any integer, $n \geq 1$.

   (b) Trace the algorithm for $n = 14$ to see it works correctly.

   (c) Prove by induction that the algorithm works correctly for any integer $n$.
   Hint: Observe that any integer $n$ always falls between two consecutive powers of 2. (For example, for $n = 14$, $2^3 < 14 < 2^4$.) In general, for every integer $n$,

   $$2^k \leq n < 2^{k+1}$$

   for some integer $k$. This will be helpful in your induction proof.

12. Consider a $2^n \times 2^n$ board, with one of its four quadrants missing. That is, the board consists of only three quadrants, each of size $2^{n-1} \times 2^{n-1}$. Let's call such a board a quad-deficient board. For $n = 1$, such a board becomes an L-shape 3-cell piece called a **tromino**, as shown below.

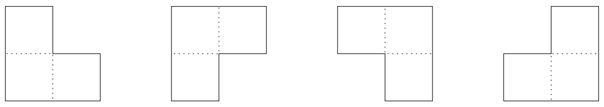


Figure 1: A Tromino, with 4 possible rotational positions.

(a) Use a **divide-and-conquer** technique to prove by induction that a quad-deficient board of size $2^n \times 2^n$, $n \geq 1$ can always be **covered** using some number of trominoes. (By covering we mean that every cell of the board must be covered by a tromino piece, and the pieces must not overlap.) Use a diagram to help describing your algorithm and the proof.

(b) Illustrate the covering produced by the algorithm for $n = 3$ (that is, $2^3 \times 2^3$ board).

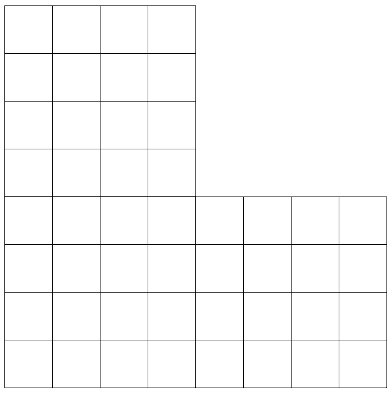


Figure 2: An $8 \times 8$ quad-deficient board.

(c) Let $f(n)$ be the total number of trominoes used to cover a $2^n \times 2^n$ quad-deficient board. Write a recurrence for $f(n)$. Solve the recurrence by repeated substitution.

13. The Fibonacci sequence [1] is defined as $F_1 = 1$, $F_2 = 1$, and

$$F_n = F_{n-1} + F_{n-2}, \quad n \geq 3.$$

(a) Compute and tabulate $F_n$ for $n = 0$ to 12.

(b) Prove the following **lower bound** on $F_n$.

$$F_n \geq 2^{\lfloor (n-1)/2 \rfloor}, \quad n > 2.$$

**Hint:** For $n > 2$, observe that $F_{n-1} \geq F_{n-2}$. (Why?) Using this relation, obtain a simpler recurrence: $F_n \geq 2F_{n-2}, \ n > 2$. Then apply repeated substitution.

(c) Prove the following **upper bound** for $F_n$.

$$F_n \leq 2^{n-2}, \quad n > 2.$$

**Hint:** Again, use the relation $F_{n-2} \leq F_{n-1}$ to obtain a simpler recurrence: $F_n \leq 2F_{n-1}, \ n > 2$. Then apply repeated substitution.

14. (a) Prove (without use of calculus) that

$$\log(n!) = \Theta(n \log n).$$

**Hints:**

- First observe that $\log(n!) = \sum_{i=1}^{n} \log i$.
- Then prove the upper bound in a trivial way.

[1] **Historical Note:** Originally, Fibonacci came up with this recurrence to describe the population growth of rabbits! Suppose that at the beginning of the year, a farm receives one pair of newly-born rabbits, and that every month each pair which is at least two-month old gives birth to a new pair. Let $F_n$ be the number of pairs at the end of month $n$, assuming that no deaths occur. Then, the number of pairs that are born at the end of month $n$ is $F_{n-2}$, thus the recurrence $F_n = F_{n-1} + F_{n-2}$. What is the number of pairs at the end of the year?

- One way to prove the lower bound is by considering only the larger $n/2$ terms in the summation. That is, $\sum_{i=1}^{n} \log i > \sum_{i=\lceil n/2 \rceil}^{n} \log i$.

(b) Prove that

$$\sum_{i=1}^{n} (i \log i) = \Theta(n^2 \log n).$$

| CS 506, Online Foundations of CS | **Homework 5 Functions & Relations Solution** | Dr. David Nassimi |
|---|---|---|

**Study: Modules 4 (Functions) and Module 5 (Relations)**

1. Consider the following sets of ordered pairs.

   (a) $S_1 = \{(1,a),(2,b),(3,c),(4,d)\}$
   (b) $S_2 = \{(1,a),(2,a),(3,d),(4,d)\}$
   (c) $S_3 = \{(1,a),(1,b),(2,c),(2,d)\}$

   Determine if each set defines a *function* from $X = \{1,2,3,4\}$ to $Y = \{a,b,c,d\}$. If the set is a function, then:

   - Determine its domain $D$, co-domain $Y$, and range $R$.
   - Is the function one-to-one?
   - Is the function *onto*?
   - If the function is one-to-one and onto, find its inverse.

   (a) $S_1$ is a function, $D = \{1,2,3,4\}, R = Y = \{a,b,c,d\}$.
   It is one-to-one and onto.
   Inverse $= \{(a,1),(b,2),(c,3),(d,4)\}$.

   (b) $S_2$ is a function, $D = \{1,2,3,4\}, Y = \{a,b,c,d\}, R = \{a,d\}$.
   It is NOT one-to-one, since $f(1) = f(2) = a$.
   It is NOT onto, since $R \subset Y$.

   (c) $S_3$ is NOT a function, since $1 \in X$ is assigned two values $a \in Y$ and $b \in Y$.

2. The functions $f(n) = n^2$ and $g(n) = 2n$ are defined on the set of positive real numbers.

   (a) Find the composition $f \circ g$. (The composition $(f \circ g)(n)$ is defined as $f(g(n))$.)
   (b) Find the composition $g \circ f$.
   (c) Find the inverse functions $f^{-1}$ and $g^{-1}$.

   (a) $(f \circ g)(n) = f(g(n)) = f(2n) = (2n)^2 = 4n^2$.
   (b) $(g \circ f)(n) = g(f(n)) = g(n^2) = 2(n^2)$
   (c) $f^{-1}(n) = \sqrt{n}$,
   $g^{-1}(n) = n/2$.

3. A sequence $(F_1, F_2, F_3, \cdots)$ is defined recursively as follows. (This recursive definition is called a *recurrence equation.*)

$$F_n = \begin{cases} 1, & n = 1, \\ 2F_{n-1} + 1, & n \geq 2. \end{cases}$$

(a) Compute $F_1, F_2, \cdots, F_{10}$ and tabulate results.

(b) Prove by induction on $n$ that

$$F_n = 2^n - 1, \quad n \geq 1.$$

(a)

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_n$ | 1 | 3 | 7 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 |

(b) Base: For $n = 1$, we see that $F_1 = 1 = 2^1 - 1$ is correct.
Hypothesis: For any $n \geq 2$, suppose that

$$F_{n-1} = 2^{n-1} - 1.$$

Then,

$$\begin{aligned} F_n &= 2F_{n-1} + 1 && \text{(from the recurrence)} \\ &= 2(2^{n-1} - 1) + 1 && \text{(from the hypothesis)} \\ &= 2^n - 2 + 1 \\ &= 2^n - 1. \end{aligned}$$

4. The Fibonacci sequence is defined recursively as follows:

$$F_n = \begin{cases} 1, & n = 1, \\ 1, & n = 2, \\ F_{n-1} + F_{n-2}, & n \geq 3. \end{cases}$$

(a) Compute and tabulate $(F_1, F_2, \cdots, F_{12)}$.

(b) Prove by induction that for all $n \geq 1$,

$$F_n \leq (1.62)^{n-1}.$$

(c) Prove by induction that for all $n \geq 1$,

$$F_n \geq (1.61)^{n-2}.$$

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $F_n$ | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 | 89 | 144 |

**Proof of the upper bound:** The two base cases are seen to be correct:

$$F_1 = 1 \leq 1.62^0$$

$$F_2 = 1 \leq 1.62^1$$

To prove for $n \geq 3$, suppose the upper bound is correct for $n-1$ and $n-2$. Then,

$$\begin{aligned} F_n &= F_{n-1} + F_{n-2} \\ &\leq 1.62^{n-2} + 1.62^{n-3} \\ &\leq 1.62^{n-1}(1/1.62 + 1/(1.62)^2) \\ &\leq 1.62^{n-1}(2.62/2.6244) \\ &\leq 1.62^{n-1} \end{aligned}$$

**Proof of lower bound:** The two base cases are seen to be correct:

$$F_1 = 1 \geq 1.61^{-1} = 1/1.61$$

$$F_2 = 1 \geq 1.61^0$$

To prove for $n \geq 3$, suppose the lower bound is correct for $n-1$ and $n-2$. Then,

$$\begin{aligned} F_n &= F_{n-1} + F_{n-2} \\ &\geq 1.61^{n-3} + 1.61^{n-4} \\ &\geq 1.61^{n-2}(1/1.61 + 1/(1.61)^2) \\ &\geq 1.61^{n-2}(2.61/2.5921) \\ &\geq 1.61^{n-2} \end{aligned}$$

5. Let $\{F_n\}$ be the Fibonacci sequence (defined above). Prove by induction that for all $n \geq 1$,

$$S(n) = \sum_{k=1}^{n} F_k = F_{n+2} - 1.$$

For the base, $n = 1$, $S(1) = F_1 = 1$ and $F_3 - 1 = 2 - 1 = 1$, so base is correct. Next, suppose it is true for some $n \geq 1$,

$$S(n) = \sum_{k=1}^{n} F_k = F_{n+2} - 1.$$

Then we will prove it is also true for $n + 1$.

$$S(n+1) = \sum_{k=1}^{n+1} F_k = S(n) + F_{n+1} = (F_{n+2} - 1) + F_{n+1} = (F_{n+2} + F_{n+1}) - 1 = F_{n+3} - 1.$$

6. Consider the following relations.

   (a) $R_1 = \{(1,1),(1,2),(2,1),(2,2),(3,3)\}$

   (b) $R_2 = \{(x,y) \mid x,y$ are positve integers $\leq 3,$ and $x+y \leq 5\}$

   (c) $R_3 = \{(x,y) \mid x,y$ are positve integers $\leq 3,$ and $x \leq y \leq x+2\}$

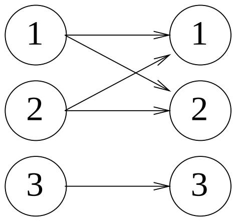
   For each relation,

   - Show the relation as a *set of ordered pairs* (if not in that form already), in *table* form, and in the form of a *directed graph.* (Show the graph with two columns of vertices, where the domain is the left column and the range is the right column.)
   - Determine whether the relation is *reflexive, symmetric, antisymmetric, transitive.* (Justify your answers.)
   - Determine if the relation is a *partial order* or an *equivalence relation.* If the latter is true, then specify the equivalence classes.

**Solution:**

(a) $R_1 = \{(1,1),(1,2),(2,1),(2,2),(3,3)\}$

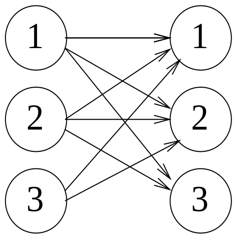
| | |
|---|---|
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |
| 2 | 2 |
| 3 | 3 |



| | |
|---|---|
| Reflexive: | Yes. For all $i$, $(i,i) \in R$. |
| Symmetric: | Yes, (1,2) and (2,1) are both in $R$. |
| Antisymmetric: | No. |
| Transitive: | Yes. |
| Partial Order: | No, it is not antisymmetric. |
| Equiv Relation: | Yes, because it is ref, symm, and trans. |
| | Equiv classes: $\{1,2\},\{3\}$. |

(b) $R_2 = \{(x,y) \mid x,y$ are positve integers $\leq 3,$ and $x+y \leq 5\}$.
Note: Positive integers are $\geq 1$ and do not include 0.

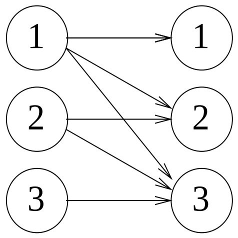
R = { (1,1), (1,2), (1,3),
(2,1), (2,2), (2,3),
(3,1), (3,2) }



| | |
|---|---|
| Reflexive: | No, $(3,3) \notin R$. |
| Symmetric: | Yes. |
| Antisymmetric: | No. |
| Transitive: | No, $(3,2),(2,3)$ in $R$, but $(3,3) \notin R$. |
| Partial Order: | No, not reflexive. |
| Equiv Relation: | No, not reflexive. |

(c) $R_3 = \{(x,y) \mid x,y$ are positve integers $\leq 3$, and $x \leq y \leq x+2\}$.
Note: Positive integers are $\geq 1$ and do not include 0.

R = { (1,1), (1,2), (1,3),
(2,1), (2,3),
(3,3) }



| | |
|---|---|
| Reflexive: | Yes. |
| Symmetric: | No, $(1,2) \in R$ but $(2,1) \notin R$. |
| Antisymmetric: | Yes. If $(x,y) \in R$, $x < y$, then $(y,x) \notin R$. |
| Transitive: | Yes. |
| Partial Order: | Yes, it is reflexive, antisymmetric, and transitive. |
| Equiv Relation: | No. |

□

7. For each of the following relations, show the Boolean matrix $A$ and compute the Boolean matrix $A^2$. Then, by comparing $A$ and $A^2$, determine whether the relation is transitive, and justify your answer.

(a) $R_1 = \{(1,1),(1,3),(2,2),(3,1),(3,3)\}$

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \qquad A^2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \text{x} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Transitive because $A^2 = A$.

(b) $R_2 = \{(1,1),(2,1),(2,3),(3,1)\}$

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad A^2 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \text{x} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Transitive because for every $i,j$, if $A^2[i,j] \neq 0$ then $A[i,j] \neq 0$ .

(c) $R_3 = \{(1,1),(2,3),(3,1)\}$

This realtion is *not* transitive, since $A^2[2,1] = 1$ but $A[2,1] = 0$.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad A\^{}2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \text{ x } \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

# Additional Exercises (Not to be handed-in)

8. Let $\{F_n\}$ be the Fibonacci sequence (defined above). Prove by induction that for all $n \geq 6$,

$$\frac{1}{\sqrt{5}} 1.61^n < F_n < \frac{1}{\sqrt{5}} 1.62^n.$$

(Hint: You need to use $n = 6$ and $n = 7$ for base cases.)

It is interesting to see how the above lower and upper bounds were determined. In Chapter 7, we will see how to find the exact solution of the Fibonacci sequence. The exact solution is

$$F_n = \frac{1}{\sqrt{5}}(r_1)^n - \frac{1}{\sqrt{5}}(r_2)^n$$

where $r_1 = \frac{1+\sqrt{5}}{2} \approx 1.618034$ and $r_2 = \frac{1-\sqrt{5}}{2} \approx -0.618034$. The second term, $(r_2)^n$ becomes a very small fraction as $n$ gets large,

$$|(r_2)^n| < 0.0557, \quad n \geq 6$$

and so $F_n \approx \frac{1}{\sqrt{5}} 1.618^n$.

9. Let $\{F_n\}$ be the Fibonacci sequence (defined above). Prove by induction that for all $n \geq 10$,

$$\frac{1}{\sqrt{5}} 1.618^n < F_n < \frac{1}{\sqrt{5}} 1.6181^n.$$

(Hint: You need to use $n = 10$ and $n = 11$ for base cases.) Note that $(1.6180)^2 = 2.6179$ and $(1.6181)^2 = 2.6182$.

10. Consider the relational database defined by the tables on p. 177 in our textbook (Edition 7).

- Table 3.6.4: EMPLOYEE
- Table 3.6.5: DEPARTMENT
- Table 3.6.6: SUPPLIER
- Table 3.6.7: BUYER

(Note: In the older edition of the textbook, these are Tables 3.4.4–3.4.7, p. 141.)

Write a sequence of database operations for each of the following queries. Also, provide the answer to each query.

(a) Find the names of all managers.

DEPARTMENT [Manager]

| |
|---|
| Jones |
| Yu |
| Zamora |
| Washington |

(b) Find the names of all employees managed by Jones.

```
T=EMPLOYEE [Manager=Jones]
T [Name]
```

| |
|---|
| Kaminski |
| Schmidt |
| Manacotti |

(c) Find all part numbers supplied by department 23.

```
T := SUPPLIER [Dept=23]
T [Part No]
```

| |
|---|
| 2A |
| 772 |

(d) Find all buyers supplied by department 23.

```
T1 := SUPPLIER [Dept=23]                Select
T2 := T1 [Part No = Part No] BUYER      Join
T2 [Name]                               Project
```

| |
|---|
| United Supplies |
| JCN Electronics |
| ABC Unlimited |

(e) Find all buyers supplied by the department managed by Jones.

```
T1 := DEPARTMENT [Manager=Jones]        Select
T2 := T1 [Dept=Dept] SUPPLIER           Join
T3 := T2 [Part No = Part No] BUYER      Join
T3[Name]                                Project
```

| |
|---|
| United Supplies |
| JCN Electronics |
| ABC Unlimited |

11. Prove each of the following simple equalities.
**Hint:** If $n$ is divisible by 4 (i.e, $n = 4k$ for some integer $k$), these are obviously correct. When $n$ is not divisible by 4, then $n = 4k + r$, where $r$ is the remainder, $r \in \{1, 2, 3\}$. Prove each equality by considering each $r$ value. (This proof method is called proof by enumeration.)

(a) $\lfloor \lfloor n/2 \rfloor /2 \rfloor = \lfloor n/4 \rfloor$

(b) $\lceil \lceil n/2 \rceil /2 \rceil = \lceil n/4 \rceil$

(c) For yourself (not to hand in this part), figure out the hybrid forms:

- $\lceil \lfloor n/2 \rfloor /2 \rceil$
- $\lfloor \lceil n/2 \rceil /2 \rfloor$

Show both of these become either $\lfloor n/4 \rfloor$ or $\lceil n/4 \rceil$.

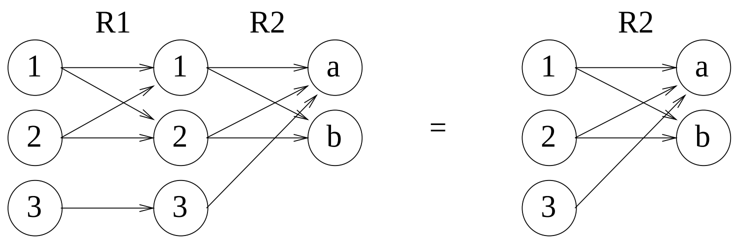
12. Let $R_1$ be a relation from $X$ to $Y$ and $R_2$ be a relation from $Y$ to $Z$. The *composition* of $R_1$ and $R_2$, denoted as $R_2 \circ R_1$, is a relation from $X$ to $Z$ defined by

$$R_2 \circ R_1 = \{(x, z) \mid (x, y) \in R_1 \text{ and } (y, z) \in R_2 \text{ for some } y \in Y\}.$$

(a) For the following relations $R_1$ and $R_2$, find $R = R_2 \circ R_1$. Show the relation $R$ as a set of ordered pairs, in table form, and in graph form.

$$R_1 = \{(1,1),(1,2),(2,1),(2,2),(3,3)\}$$
$$R_2 = \{(1,a),(1,b),(2,a),(2,b),(3,a)\}$$

**Solution:** The composition $R_2 \circ R_1$ becomes the same as $R_2$ in this case. $\square$



(b) Show the Boolean matrices $A_1$ and $A_2$ for $R_1$ and $R_2$, respectively. (The matrices are $3 \times 3$ and $3 \times 2$, respectively. For $R_2$, columns 1 and 2 should be labeled $a$ and $b$ respectively.)

Then, compute the Boolean product $A_1 \times A_2$. (The product matrix is also Boolean, with all entries 0 or 1.) Verify that the product matrix correctly represents the relation $R$.

$$\text{A1} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \text{A2} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{A 1 x A2} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

13. Let $R$ be a relation and $A$ be its Boolean matrix. Let $A^2 = A \times A$ be the Boolean product, where all entries in the product matrix are either 0 or 1. That is,

$$A^2[i,j] = \bigvee_{k=1}^{n} (A[i,k] \wedge A[k,j]).$$

(a) Prove that $R$ is transitive **<u>if and only if</u>**

$$\forall i,j, \quad (A^2[i,j] = 1) \rightarrow (A[i,j] = 1).$$

(This implication means that whenever $A^2[i,j]$ is nonzero, $A[i,j]$ is also nonzero.)

**Hint:** You may provide the proof by the following two steps:

i. Prove that if $R$ is transitive, and $A^2[i,j] = 1$ for any $i,j$, then $A[i,j] = 1$.

ii. Prove that if $R$ is not transitive, then $\exists i,j, \quad A^2[i,j] = 1$ and $A[i,j] = 0$.

(b) Prove that if $R$ is reflexive, then

$$\forall i,j, \quad (A[i,j] = 1) \rightarrow (A^2[i,j] = 1).$$

(This implication means that whenever $A[i,j]$ is nonzero, $A^2[i,j]$ is also nonzero.)

(c) Let $R$ be a reflexive relation. Prove that $R$ is transitive **<u>if and only if</u>**

$$A^2 = A.$$

| CS 506, Online<br>Foundations of CS | **Homework 9**<br>**Discrete Probability**<br>**Solution** | Dr. David Nassimi |
|---|---|---|

**Study Module 10: Discrete Probability, Lottery and Games**

1. A random flip of a coin gives either a "head" or "tail", with equal probabilities. Consider an experiment where we repeatedly flip a coin until we get a head. What is the probability that the first head occurs on $k$th flip of the coin? Explain.

The first $k-1$ flips must be tail AND the $k$th flip must be head. So by the product rule, the probability is

$$\frac{1}{2} \times \frac{1}{2} \times \cdots \frac{1}{2} \times \frac{1}{2} = \frac{1}{2^k}.$$

2. A pair of 6-sided dice is rolled. Derive the probability that at least one of them is 5. Find the probability in three ways:

   (a) List all possible cases, where at least one dice is 5.

   (b) Use the formula,
   $$P(A \vee B) = P(A) + P(B) - P(A \wedge B).$$

   (c) Use the formula,
   $$P(A \vee B) = 1 - P(\neg A \wedge \neg B).$$

**Solution:**

(a) We enumerate all cases where one of them is 5, which are 11 cases. (See below.) So, the probability is 11/36.

| $D1D2$ |
|---|
| 51 |
| 52 |
| 53 |
| 54 |
| 55 |
| 56 |
| 15 |
| 25 |
| 35 |
| 45 |
| 65 |

(b)

$$\begin{aligned} P(D1 = 5 \vee D2 = 5) &= P(D1 = 5) + P(D2 = 5) - P(D1 = 5 \wedge D2 = 5) \\ &= 1/6 + 1/6 - 1/36 = 11/36. \end{aligned}$$

(c)

$$\begin{aligned} P(D1 = 5 \vee D2 = 5) &= 1 - P(D1 \neq 5 \wedge D2 \neq 5) \\ &= 1 - P(D1 \neq 5) * P(D2 \neq 5) \\ &= 1 - \frac{5}{6} * \frac{5}{6} \\ &= 1 - 25/36 = 11/36. \end{aligned}$$

□

3. A pair of dice is thrown. Compute the probability that their sum equals to $t$ for each $t$ from 2 to 12, and tabulate them.

**Solution:**

| $t$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Number of ways for sum=t | 1 | 2 | 3 | 4 | 5 | 6 | 5 | 4 | 3 | 2 | 1 |
| Probability | 1/36 | 2/36 | 3/36 | 4/36 | 5/36 | 6/36 | 5/36 | 4/36 | 3/36 | 2/36 | 1/36 |

For example, for $sum = 7$, we list the possible cases, which are 6 cases. So $P(sum = 7) = 6/36$.

|  |
|---|
| 16 |
| 25 |
| 34 |
| 43 |
| 52 |
| 61 |

□

4. A pair of dice is rolled. Find the probability for each of the following events.

(a) The first dice is $\geq 4$.
(b) At least one of them is $\geq 4$.
(c) The sum is $\geq 8$.
(d) The sum is $\geq 8$, given that the first dice is $\geq 4$.
(e) The sum is $\geq 8$, given that at least one of them is $\geq 4$.
(f) At least one of them is $\geq 4$, given that the sum is $\geq 8$.

**Solution:**

(a) The first dice is $\geq 4$.
This probability is $3/6 = 1/2$.

(b) At least one of them is $\geq 4$.
One way is to use the formula:

$$\begin{aligned} P((D1 \geq 4) \vee (D2 \geq 4)) &= P(D1 \geq 4) + P(D2 \geq 4) - P((D1 \geq 4) \wedge (D2 \geq 4)) \\ &= 1/2 + 1/2 - (1/2) * (1/2) = 3/4. \end{aligned}$$

This may also be found by listing all cases, which are 27. So, the probability becomes $27/36 = 3/4$.

(c) The sum is $\geq 8$.

$$\begin{aligned} P(sum \geq 8) &= P(sum = 8) + P(sum = 9) + P(sum = 10) + P(sum = 11) + P(sum = 12) \\ &= (5+4+3+2+1)/36 = 15/36. \end{aligned}$$

(d) The sum is $\geq 8$, given that the first dice is $\geq 4$.
List all entries with $D1 \geq 4$, which are 18. Then mark those entries with $sum \geq 8$, marked by *. The probability is the ratio of the number of entries with $sum \geq 8$ (marked by *) out of the total number of entries with $D1 \geq 4$. So the probability becomes

$$(3+4+5)/18 = 12/18 = 2/3.$$

| | |
|---|---|
| 41 | |
| 42 | |
| 43 | |
| 44 | $*$ |
| 45 | $*$ |
| 46 | $*$ |
| 51 | |
| 52 | |
| 53 | $*$ |
| 54 | $*$ |
| 55 | $*$ |
| 56 | $*$ |
| 61 | |
| 62 | $*$ |
| 63 | $*$ |
| 64 | $*$ |
| 65 | $*$ |
| 66 | $*$ |

(e) The sum is $\geq 8$, given that at least one of them is $\geq 4$.
List all cases with at least one dice $\geq 4$. (There are 27 cases). Then mark those entries with $sum \geq 8$. (There are 15 cases, marked by *.) So the probbility is the ratio of 15 over 27.

$$P = 15/27 = 5/9.$$

| | |
|---|---|
| 41 | |
| 42 | |
| 43 | |
| 44 | $*$ |
| 45 | $*$ |
| 46 | $*$ |
| 51 | |
| 52 | |
| 53 | $*$ |
| 54 | $*$ |
| 55 | $*$ |
| 56 | $*$ |
| 61 | |
| 62 | $*$ |
| 63 | $*$ |
| 64 | $*$ |
| 65 | $*$ |
| 66 | $*$ |
| 14 | |
| 24 | |
| 34 | |
| 15 | |
| 25 | |
| 35 | $*$ |
| 16 | |
| 26 | $*$ |
| 36 | $*$ |

(f) At least one of them is $\geq 4$, given that the sum is $\geq 8$.
Given that the sum is $\geq 8$, then at least one dice must be $\geq 4$. (Otherwise, if both dice are $< 4$ then the sum is $< 8$.) So this probabilty is 1.

□

5. Jersey-Cash-5 is a New Jersey Lottery that "benefits education and institutions". In this lottery, a game is played by picking 5 distinct numbers out of numbers 1 to 40.

Compute the number of ways and the probability of winning each of the following prizes. Show the combinatorial derivation. Use a calculator to perform the final step in the computation. Express the final result as a ratio $1/x$ where the denomintor $x$ is rounded to the nearest integer.

(a) Jackpot: Match 5 out of 5.
(b) Second Prize: Match 4 out of 5. (You win \$500.)
(c) Third Prize: Match 3 out of 5. (You win \$11.)

**Solution:**

(a) The sample size (number of all possible tickets) is

$$U = C(40,5) = (40*39*38*37*36)/(5*4*3*2*1) = 658,008.$$

So the probability of winning the jackpot is

$$1/U = 1/658008.$$

(b) Second Prize: Match 4 out of 5.
The number of ways to win this prize is:

$$W2 = C(5,4) * C(35,1) = 5 * 35 = 175.$$

The probability is

$$P2 = W2/U = 175/658008 \approx 1/3760.$$

(c) Third Prize: Match 3 out of 5.

$$W3 = C(5,3) * C(35,2) = 10 * 595 = 5950.$$

$$P3 = 5950/U = 5950/658008 \approx 1/110.$$

□

6. A poker hand is a random drawing of 5 cards out of a deck of 52 cards (with no joker). The cards have 4 **suits** (Diamond, Hearts, Spade, Club) and 13 **kinds**: (ace, king, queen, jack, 10, 9, 8, 7, 6, 5, 4, 3, 2), where an ace counts as either the highest or lowest. (For the list of poker hands and illustrations, please see wikipedia.org.)

For the purpose of this problem, the poker hands are listed below in ranked order, along with the definition of each hand. Each hand is defined so as not to include any higher hand(s). For example, a *straight* hand means five cards in sequence, but not all in the same suite (otherwise, it would become straight flush).

Derive the number of ways $W()$, and the probability $P()$, for each poker hand. Show the combinatorial derivation. Find each probabilty in the form of a ratio $1/x$, where $x$ is the nearest integer.

(a) Straight Flush: Five cards of the same suit and in consecutive order.

The number of possible poker hands, also known as the sample size, is

$$U = C(52,5) = (52 * 51 * 50 * 49 * 48)/(5 * 4 * 3 * 2 * 1) = 2,598,960.$$

The high card of a straight flush has 10 possible values, since it may be one of

$$\{Ace, king, queen, jack, 10, 9, 8, 7, 6, 5\}$$

but not 4,3,2. And there are 4 suits. So,

$$\begin{aligned} W &= 10 * 4 = 40. \\ P &= W/U = 40/2598960 \approx 1/64974. \end{aligned}$$

(b) Four-of-a-Kind (4,1): Four of one kind and one of another kind.

$$\begin{aligned} W &= C(13,1) * C(4,4) * C(12,1) * C(4,1) \\ &= 13 * 1 * 12 * 4 = 624 \\ P &= 624/U \approx 1/4165. \end{aligned}$$

(c) Full House (3,2): Three of one kind and two of another kind.

$$\boxed{\begin{aligned} W &= C(13,1) * C(4,3) * C(12,1) * C(4,2) \\ &= 13 * 4 * 12 * 6 = 3744 \\ P &= 3744/U \approx 1/694. \end{aligned}}$$

(d) Flush: Five cards of the same suite (but not in sequence).

$$\boxed{\begin{aligned} W &= C(4,1) * [C(13,5) - 10] \\ &= 4 * [(13 * 12 * 11 * 10 * 9)/(5 * 4 * 3 * 2 * 1) - 10] \\ &= 4 * 1277 = 5108 \\ P &= 5108/U \approx 1/509. \end{aligned}}$$

(e) Straight: Five cards in sequence (but not all the same suite).

$$\boxed{\begin{aligned} W &= C(10,1) * [C(4,1)^5 - 4] \\ &= 10 * (4^5 - 4) = 10200 \\ P &= 10200/U \approx 1/255. \end{aligned}}$$

(f) Three-of-a-Kind (3,1,1): Three of one kind, one of another kind, and one of a third kind.

$$\boxed{\begin{aligned} W &= C(13,1) * C(4,3) * C(12,2) * C(4,1) * C(4,1) \\ &= 13 * 4 * 66 * 4 * 4 = 54912 \\ P &= 54912/U \approx 1/47. \end{aligned}}$$

(g) Two Pairs (2,2,1): Two of one kind, two of another kind, and one of a third kind.

$$\boxed{\begin{aligned} W &= C(13,2) * C(4,2) * C(4,2) * C(11,1) * C(4,1) \\ &= 78 * 6 * 6 * 44 \\ &= 123,552. \\ P &= 123552/U \approx 1/21. \end{aligned}}$$

(h) One Pair (2,1,1,1): Two of one kind, and three other kinds, with one card from each kind.

$$\begin{aligned} W &= C(13,1) * C(4,2) * C(12,3) * C(4,1) * C(4,1) * C(4,1) \\ &= 13 * 6 * 220 * 4^3 \\ &= 1,098,240. \\ P &= 1098240/U \approx 1/2.36 \end{aligned}$$

(i) High Card: This occurs when none of the above hands is made. (That is, the 5 cards are 5 different kinds but not all in sequence, and not all the same suit.)

$$\begin{aligned} W &= [C(13,5) - 10] * [C(4,1)^5 - 4] \\ &= 1277 * 1020 \\ &= 1,302,540. \\ P &= 1302540/U \approx 1/1.99 \end{aligned}$$

Check your work by adding the number of ways for all 9 cases. The total must be $C(52,5)$. For this check, you need to compute the number of ways to get the last hand (high card) dirctly, which is:

$$[C(13,5) - 10] * [(C(4,1))^5 - 4] = 1277 * 1020 = 1,302,540.$$

Check:

| Hand | Number of Ways |
|---|---|
| Straight Flush | 40 |
| 4-of-a-kind | 624 |
| Full House | 3744 |
| Flush | 5108 |
| Straight | 10200 |
| 3-of-a-kind | 54912 |
| 2-pair | 123552 |
| 1-pair | 1098240 |
| High-card | 1302540 |
| Total: | 2598960 |

7. Consider a poker hand dealt with the first two cards facing up, and the remaining 3 cards face down. Compute the conditional probability of three-of-a-kind (3,1,1), given that the first two face-up cards are (Ace of hearts, King of Diamond).

Note that the toal number of possible poker hands, given that the first two cards are (ace,king) is

$$U = C(50,3) = (50 * 49 * 48)/(3 * 2 * 1) = 19600.$$

Hint: There are three ways of ending up with a 3-of-a-kind hand, given that the first two cards are (Ace,King):

- Get two more aces, and one card of a third kind, OR

- Get two more kings, and one card of a third kind, OR
- Stay with one ace and one king, and get 3 cards of a third kind.

**Correct Solution:** As stated in the hint, there are three possible ways of getting 3-of-a-kind, after the first two cards, which are (ace,king). Let us find the number of ways for each possible case.

- Get two more aces, and one card of a third kind.
  The number of ways is:
  $$W_1 = C(3, 2) * C(11, 1) * C(4, 1) = 3 * 11 * 4 = 132$$
- Get two more kings, and one card of a third kind.
  The number of ways for this is the same as above, which is
  $$W_2 = C(3, 2) * C(11, 1) * C(4, 1) = 3 * 11 * 4 = 132$$
- Stay with one ace and one king, and get 3 cards of a third kind.
  The number of ways is:
  $$W_3 = C(11, 1) * C(4, 3) = 11 * 4 = 44$$

By the addition principle, the total number of ways is the sum of the three:

$$W = W_1 + W_2 + W_3 = 132 + 132 + 44 = 308.$$

And we know the total number of poker hands after the first two cards is:

$$U = C(50, 3) = (50 * 49 * 48)/(3 * 2 * 1) = 19600$$

Therefore, the conditional probability becomes:

$$P = W/U = 308/19600 \approx 1/64.$$

# Additional Exercises (Not to be handed-in)

8. Let us look at the last problem again. We want to compute the conditional probability of 3-of-a-kind (3, 1, 1) given that the first two cards are an ace and a king.

   **A wrong counting method (Warning to avoid a common mistake):**
   Suppose a student **erronously** performs this computation as follows: Get one card from a third kind, so you have (Ace, king, and one of another kind). Then get two more cards from one of the three kinds! So, in this case,

   $$W = C(11, 1) * C(4, 1) * C(3, 1) * C(3, 2) = 11 * 4 * 3 * 3 = 396$$
   $$P = W/U = 396/19600 \approx 1/49$$

Why is this wrong?!

The number of ways to get 3 of a third kind must be an *unordered* drawing of 3 cards out of 4. But in the above computation, the drawing is wrongly *ordered.* That is, we first draw a card from another kind, and then draw two more cards from that kind, which results in wrong count!

9. Consider a poker hand dealt with the **first two cards facing up**, and the remaining 3 cards face down. Compute the number of ways and the probability for each of the poker hands, given that the first two face-up cards are two aces.

   Check your work by adding the number of ways for all cases. The total must be $C(50, 3)$.

10. Consider a poker hand dealt with the **first two cards facing up**, and the remaining 3 cards face down. Compute the number of ways and the probability for each of the poker hands, given that the first two face-up cards are (Ace, King).

    Check your work by adding the number of ways for all cases. The total must be $C(50, 3)$.

11. Consider a deck of 53 cards, which includes a joker (wild card). The possible poker hands are:

    (a) Five-of-a-kind (Four of one kind, plus a joker)
    (b) Straight Flush (five cards same suit and in consecutive order).
    (c) Four-of-a-Kind (4,1)
    (d) Full House (3,2)
    (e) Flush
    (f) Straight
    (g) Three-of-a-Kind (3,1,1)
    (h) Two Pairs (2,2,1)
    (i) One Pair (2,1,1,1)
    (j) High Card: This occurs when none of the above hands is made.

    For example, a Four-of-a-Kind hand can be made in two ways:

    - Without joker: four of one kind plus one of another kind.
    - With a joker: three of one kind, plus a joder, plus one of another kind.

    Compute the number of ways to get each hand, and the probability.

    Check your work by adding the number of ways for all cases. The total must be $C(53, 5)$.