**lexicographical order:** In alphabetical order.

**graph traversal:** some method that touches every vertex of a graph.

**computational complexity:** The time it takes to complete an algorithm according n value.

**group of units mod n:** All possible numbers mod n

**greedy strategy:** trying to grab branch with the least weight on a minimal spanning tree.

**euler phi function:** # elements in Zn

**inverse of k:** k^(p-2) mod p = inverse of k mod p

**Big Oh of n:** O(n)

**Adjacency List:** Represents a graph of unordered list

**Adjacency Matrix:** Represents vertices of a graph.

**Zn:** the set containing the numbers {0,.....,n-1} and arithmetic mod n.

**Group of unites:** elements in Zn coprime to n.

**Inverse of k in group of units mod n:** the term we multiply k by to get 1.

**GCD:** Greatest common divisor

**Order of an element mod n:** smallest t s.t. a^t congruent to 1 mod n

**Fermats Little Theorem:** if n is a prime, a is in Zn then a^(n-1) congruent to 1 mod n

**Euler Theorem:** if a is in Zn then a^(phi of n) congruent mod n.

**Pseudoprime:** “not quite real” prime

**1. Given a list of integers, show how they are sorted by quicksort.**

**2. given a permutation, compute its rank.**

**3. given a positive integer compute a permutation with this rank.**

**4. how can we use the unrank algorithm to list all the permutations in lex order.**

**5. describe the strategy of the rabin/karp algorithm for pattern matching**

Takes a string pattern p and hashes it, the hashed string to compare it to. then it compares the number s and moves down the string until a match is found or hits the end of the string As its moving p it subtracts the largest integer value and adding the new end value.

**6. given a directed graph, show a topological fort on the vertices.**

**7. given a weighted graph, show a minimal spanning tree.**

**7.5. describe prim’s algorithm for finding the minimal spanning tree.**

**8. suppose you counted the number of operations required for your algorithm and go A\*n^3. if the**

**time it takes on your laptop is T, how long will it take to double the size of the input? so what is the complexity of the algorithm? what if your algorithm took time A\*2^n?**

**9. give an example of a well known algorithm for each of the common complexity classes.**

O(1) = Table lookup O(n^2) = Matrix Addition O(n^3) = Matrix Multiplication

O(log n) = BST O(2^n) = Gen all Subsets O(n!) = Gen All Permutations

O(n log n) = RST Sort O(log2n) = Mod Exponentiation = we convert it to binary

1. give the code for a method that, given positive integer n, computes a random permutation of {0,1,....,n-1}.

**public static int[] RandomPermutation(int n){  
   
 int[] array = new int[n];  
 for(int i = 0; i < array.length; i++)  
 array[i] = i;  
  
 for(int i = 0; i < n; i++){  
  
 Random rand = new Random();  
   
 int ran = i + rand.nextInt (n-i);  
  
   
 int temp = array[i];  
 array[i] = array[ran];  
 array[ran] = temp;  
 }   
 return array;  
 }**

2. give the psudocode for modular exponentiation (or the java code).

**public static int ModularExp(int a, int b, int n){  
 int d = 1;  
 String bin = Integer.toBinaryString(b);  
 for(int i = 0; i < bin.length(); i++){  
 d = (d\*d)%n;  
 if(bin.charAt(i) == '1'){  
 d = (d\*a)%n;  
 }  
 }  
 return d;  
 }**

3. give the psudocode or the java code for computing all subsets of {0,1,..., n-1}.

**public static int euclid(int n, int m){  
 if(m == 0){  
 return n;  
 }  
 else{  
 return euclid(m, n%m);  
 }  
 }**

RSA Encryption:

1)public key scheme

publish(e,n), n=pq, 2 big primes

e must be large ~ p or q

GCD(e, phi(p\*q))

2) take plaintext and translate each character to ascii value

3) Block these such that the largest possible value is < n. this is the plain text p.

4. encyper it we take the plain text and raise it to the exponent to get the cyper text.

5) can only be decrypted by you know phi(pq)

6) decypherment compile e^-1 mod phi(pq). Cyper text ^ e-1 mod n.