CS 124 Section #8 RSA, Random Walks, Linear Programming 3/27/17

1 Extended Euclidean Algorithm

Given a, b, find x, y such that $ax + by = d$ where d is the GCD of a, b. This will be necessary in implementing RSA encryption.

Algorithm 1 Extended Euclidean Algorithm

```
procedure FINDGCD(a,b)
    $a \leftarrow$ larger number
    $b \leftarrow$ smaller number
    return $\leftarrow (d, x, y)$ such that $ax + by = d$ with $d = \gcd(a, b)$.
Base Case:
    if $b = 0$ then return $(a, 1, 0)$
Recursive Case:
    $k \leftarrow (a - a \mod b)/b$
    $(d, x, y) \leftarrow$ FINDGCD$(b, a \mod b)$
     return $(d, y, x - ky)$
```

- In each step, k is equal to $\lfloor \frac{a}{b} \rfloor$
- When we recursively find (d, x, y), this satisfies the equation

  $$b \cdot x + (a \mod b) \cdot y = d$$

  But we have that $a = kb + (a \mod b)$. Substituting, for $a \mod b$ we get:

  $$b \cdot x + y \cdot (a - kb) = d$$

  And after regrouping terms, this gives us:

  $$a \cdot y + b \cdot (x - ky) = d$$

- We will use the extended euclidean algorithm for finding *inverses* modulo p, i.e. a is the inverse of b modulo p if $ab = 1 \mod p$. This only exists if $d = \gcd(a, b) = 1$.

Exercise 1. Run the extended Euclidean algorithm on $a = 51$ and $b = 20$. Use the table to help you organize your work. *Hint:* Fill up the table for a, b, k first, then work your way up from the bottom for x and y.

a	b	k	x	y
51	20	2		

Solution
Fill up the following table, going down for a, b, and k, and going up for x and y, keeping $ax + by = \text{GCD}(a, b)$.

a	b	k	x	y
51	20	2	**11**	**-28**
20	11	1	-6	11
11	9	1	5	-6
9	2	4	-1	5
2	1	2	1	-1
1	0		1	1

At the end we have $51 \cdot 11 - 20 \cdot 28 = 1$ as desired.

2 RSA

RSA is an encryption algorithm loosely based on the assumption that factoring numbers is difficult. While this fact is unproven (we don't even know whether integer factorization is in NP, not to mention P = NP), it seems pretty secure.

The encryption scheme involves having a public key k_e and a private key k_d. When Alice wants to send a message to Bob, she encodes the message with k_e and then Bob decodes the message with k_d. We need to show that (1) It is hard to compute the private key from the public key and (2) It is hard to determine the message m from the encryption of m with the public key without having the private key.

In order to efficiently implement RSA, we need the help of a few algorithms:

- **Repeated Squaring**. This allows us to calculate exponents efficiently.
- **Extended Euclidean Algorithm**. We need this to calculate the private key exponent d from the public key exponent e.
- **Rabin-Miller Primality Test**. Allows us to probabilistically find large primes efficiently.

Creating the keys:

1. Bob finds primes p, q and then calculates $n = pq$.
2. Bob also chooses a number $e < n$ such that e is coprime with $\phi(n) = (p-1)(q-1)$, called Euler's totient function. We often will choose e to be $2^{2^4} + 1$ which is prime.
3. Bob calculates d such that $de = 1 \mod \phi(n)$. This is done using the extended euclidean algorithm where we try to find x, y such that $ex + \phi(n)y = \gcd(e, \phi(n)) = 1$, which would imply that $ex = 1 \mod \phi(n)$.
4. The public key is $k_e = (n, e)$ and the private key is $k_d = d$.

For plaintext m and ciphertext c:

Encryption. $c = m^e \mod n$.

Decryption. $c^d \mod n = m^{ed \mod \phi(n)} \mod n = m$

Exercise 2. Alice generating a new RSA key, and generates the very large, very secure primes $p = 11$ and $q = 29$. She chooses $e = 3$. What is n? What value of d should be used in the private key? What is the encryption of the message $m = 100$?

Solution
Multiply to get $n = 319$. Since e and $\phi(n)$ are coprime there exist c and d such that $c\phi(n) + de = 1$. Note that this d satisfies $ed = 1 \mod \phi(n)$. We can solve for d with the extended Euclidean algorithm to get $d = 187$ ($1 = -93 \cdot 3 + 1 \cdot 280$, $-93 \equiv 187 \mod 280$). Encryption is $100^3 \mod 319 = 254$.

3 Random Walks

A random walk is an iterative process on a set of vertices V. In each step, you move from the current vertex v_0 to each $v \in V$ with some probability. The simplest version of a random walk is a one-dimensional random walk in which the vertices are the integers, you start at 0, and at each step you either move up one (with probability 1/2) or down one (with probability 1/2).

2-SAT: In lecture, we gave the following randomized algorithm for solving 2-SAT. Start with some truth assignment, say by setting all the variables to false. Find some clause that is not yet satisfied. Randomly choose one the variables in that clause, say by flipping a coin, and change its value. Continue this process, until either all clauses are satisfied or you get tired of flipping coins.

We used a random walk with a completely reflecting boundary at 0 to model our randomized solution to 2-SAT. Fix some solution S and keep track of the number of variables k consistent with the solution S. In each step, we either increase or decrease k by one. Using this model, we showed that the expected running time of our algorithm is $O(n^2)$.

Exercise 3. This weekend, you decide to go to a casino and gamble. You start with k dollars, and you decide that if you ever have $n \geq k$ dollars, you will take your winnings and go home. Assuming that at each step you either win \$1 or lose \$1 (with equal probability), what is the probability that you instead lose all your money?

Solution
Let P_k be the probability that you win if you currently have \$$k$

Set up a recurrence relation: $P_0 = 0$, $P_n = 1$, $P_k = 1/2P_{k+1} + 1/2P_{k-1}$. It seems like P_k is the average of P_{k-1} and P_{k+1} for all $0 < k < n$, which suggests a linear function some sort. Thus, the linear function that would take $P_0 = 0$ and $P_n = 1$ is $P_k = k/n$, which we confirm to be a correct solution.

Solution: $P_k = k/n$, so you lose all your money with probability $1 - k/n$

Exercise 4. Consider the same situation as before. How many steps are expected to occur before you either run out of money or earn n and decide to leave?

Solution
Let E_k be the expected number of steps remaining if you currently have \$$k$.

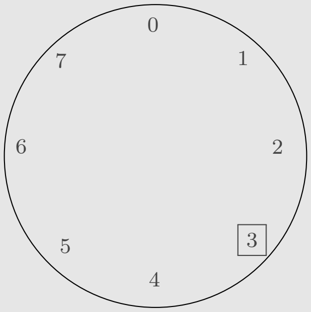
Set up a recurrence relation as in the previous problem: $E_0 = 0$, $E_n = 0$, $E_k = 1/2E_{k-1} + 1/2E_{k+1} + 1$. We guess the form of the solution to be a quadratic function, as each term is equal to the average of its two surrounding terms plus a constant. Seeing that E_k has "roots" at $k = 0$ and $k = n$, our function must be of the form $E_k = ak(n - k)$ for some constant a. As we have

$$E_1 = \frac{1}{2}E_0 + \frac{1}{2}E_2 + 1 = \frac{1}{2}E_2 + 1,$$

we have $E_1 = a(n - 1)$ and $E_2 = 2a(n - 2)$. Solving, we see that $a = 1$. Thus, we see that $E_k = k(n - k)$.

Exercise 5. (Challenging) Now suppose that there are $n + 1$ people in a circle numbered $0, 1, \ldots, n$. Person 0 starts with a bag of candy. At each step, the person with the bag of candy passes it either left or right with equal probability. The *last* person to receive the bag wins (and gets to keep all the candy). So if you were playing, then you would want to receive the bag only after everyone else has received the bag at least once. What is the probability that person i wins?

Solution
CS 124 Solution:
What needs to happen in order for person i to be the last person to receive the bag of candy? Somehow, persons $0, 1, 2, \ldots i - 1, i + 1, \ldots, n, n + 1$ must have received the bag of candy all before person i has. Let's take the example below where $n = 7$ and so there are 8 people standing in a circle.



Suppose we wanted to calculate the probability that person 3 is the last person with the candy. In order for that to happen, one of the following situations must happen:

- Person 2 receives the candy bag from person 1, passes it back to person 1, and then the bag travels all the way around the circle to person 4 who finally passes the bag to person 3.
- Person 4 receives the candy bag from person 5, who passes back to person 5, and then the bag travels all the way around the circle to person 2 who finally passes it to person 3.

What's the probability that the candy bag, which starts from person 0, will get to person 2 before it gets to person 4? We can think of that as a random walk on a line, which is the above circle "cut" at 3. We can think of this as a random walk where we start at 0 and finish when we reach either 2 or 4. $n = 6$ in our case, and k is how far away you are from the desired endpoint. By exercise 3, we know that the probability of getting to 2 before 4 is $\frac{2}{6}$ while the probability of getting to 4 before 2 is $\frac{4}{6}$.

Once the bag of candy ends up with person 2 or 4, now what has to happen is that the bag of candy must travel all the way to 3, but via the long way around. For example, suppose that the bag is with person 4 right now. It must travel to person 3 via the long way around. This is like having the following random walk:

$$3 - 4 - 5 - 6 - 7 - 0 - 1 - 2 - 3$$

where we start at 4 and try to get to the 3 on the right hand side while avoiding the 3 on the left hand side. This is a random walk with $n = 7$. We start with 1 from the left, and so the probability of us getting to the right before the left is, by exercise 3, $\frac{1}{7}$.The analogous is true if we assumed that person 2 got the candy bag before person 4.

Therefore, the two bulleted points have probability $\frac{2}{6} \cdot \frac{1}{7}$ and $\frac{4}{6} \cdot \frac{1}{7}$ respectively, and thus the probability that aperson 3 is the last to get the candy bag is $\frac{1}{7}$.

One can now generalize the argument for any starting person besides person 0 (who loses automatically) and see that each person besides person has a $\frac{1}{n}$ of taking all the candy.

Stat 110 Solution:
Note that $Pr(\text{Person 0 wins}) = 0$ (sad!). For all $i \neq 0$, we can partition the probability space to two events: That i receives the package from the left or from the right. So $\Pr(i \text{ wins}) = \Pr(i + 1 \text{ before } i - 1) \Pr(i \text{ wins} \mid i + 1 \text{ before } i - 1) + \Pr(i - 1 \text{ before } i + 1) \Pr(i \text{ wins} \mid i - 1 \text{ before } i + 1)$

Calculate $\Pr(i \text{ wins} \mid i + 1 \text{ before } i - 1)$: Consider the moment when $i + 1$ first gets the candy. This is the configuration: $i, i + 1, i + 2, \ldots, n, 0, 1, \ldots, i - 1$ where $i + 1$ has the bag of candy, and we want to find the probability that the bag of candy makes it all the way to $i - 1$ without ever touching i. Observe that this reduces to the previous problem: $\Pr(i \text{ wins} \mid i + 1 \text{ before } i - 1) = 1 - (n - 1)/n = 1/n$

By symmetry, $\Pr(i \text{ wins} \mid i - 1 \text{ before } i + 1) = 1/n$ also Therefore $\Pr(i \text{ wins}) = 1/n * (\Pr(i + 1 \text{ before } i - 1) + \Pr(i - 1 \text{ before } i + 1)) = 1/n$

4 Linear programming

Objective function (to be minimized or maximized) and constraints, all linear in some variables $x_1, \ldots, x_n$.

Geometric visualization:

- A linear equation of the form $c_1 x_1 + \ldots c_n x_n = c$ determines a hyperplane in n-dimensional space.
- A constraint $c_1 x_1 + \ldots c_n x_n \leq c$ (or $\geq c$) therefore defines a half-space, which consists of all points to one side of the hyperplane.
- The set of points satisfying all the constraints is given by the intersection of all these half-spaces, which is a convex polyhedron. (You can think about why it has to be convex.)
- The objective function, $a_1 x_1 + \ldots + a_n x_n$, can be thought of as a moveable hyperplane: We want to find the highest (maximization) or lowest (minimization) value of α such that the hyperplane $a_1 x_1 + \ldots + a_n x_n = \alpha$ still intersects the polyhedron. Therefore the optimum occurs at a corner of the polyhedron, which is why the **simplex algorithm** works.

Exercise 6. Please express the following problem as a linear programming problem: A farmer has 10 acres to plant in wheat and rye. He has to plant at least 7 acres. However, he has only $1200 to spend and each acre of wheat costs $200 to plant and each acre of rye costs $100 to plant. Moreover, the farmer has to get the planting done in 12 hours and it takes an hour to plant an acre of wheat and 2 hours to plant an acre of rye. If the profit is $500 per acre of wheat and $300 per acre of rye how many acres of each should be planted to maximize profits?

Solution
Let x be the number of acres of wheat, and y be the number of acres of rye. Then our constrains are:

$$\begin{aligned} x + y &\leq 10 \\ x + y &\geq 7 \\ 200x + 100y &\leq 1200 \\ x + 2y &\leq 12 \end{aligned}$$

and we wish to maximize $500x + 300y$.