cpt_s 350

Homework

11641327 Yu-Chieh Wang

2020/4/23

1. RSA decryption

   baby-ASCII-table

1	2	3	4	5	6	7	8	9	0
A	E	G	I	O	R	T	X	!	∅

   First of all, we translate the characters into digits.

I	T	G	!	A	A	E	X	E	X
4	7	3	9	1	1	2	8	2	8
M1									

I	R	R	G	!	I	G	R	X	I
4	6	6	3	9	4	3	6	8	4
M2									

O	I	X	G	E	R	E	A	G	O
5	4	8	3	2	6	2	1	3	5
M3									

   Then, we use python to run the RSA decryption process and get the results.

```
e = 49
n = 10539750919
M1 = 4739112828
M2 = 4663943684
M3 = 5483262135

print("C1 =", pow(M1, e) % n)
print("C2 =", pow(M2, e) % n)
print("C3 =", pow(M3, e) % n)
```

```
C1 = 3656516678
C2 = 1922627380
C3 = 359193874
```

   Finally, we translate the digits into characters.

C1									
3	6	5	6	5	1	6	6	7	8
G	R	O	R	O	A	R	R	T	X

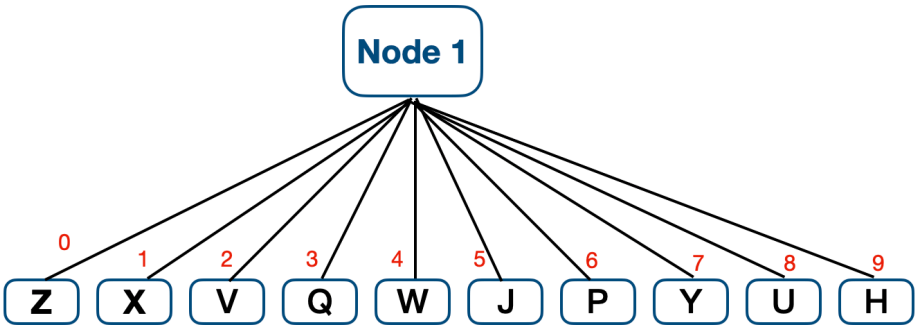
C2									
1	9	2	2	6	2	7	3	8	0
A	!	E	E	R	E	T	G	X	∅

C3									
	3	5	9	1	9	3	8	7	4
	G	O	!	A	!	G	X	T	I

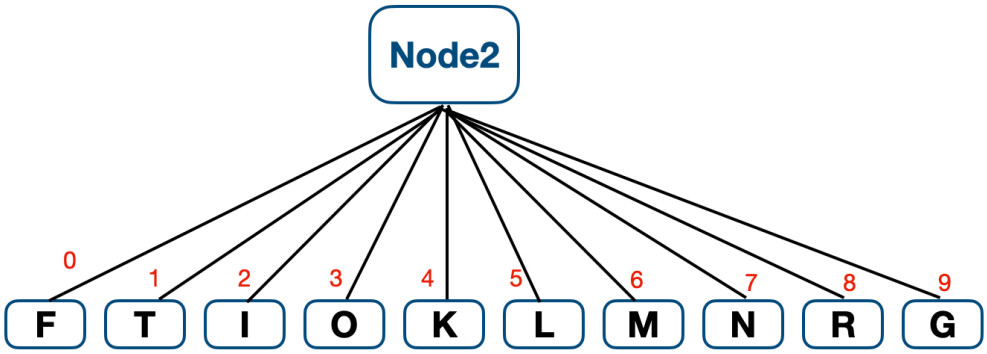
2. Design new Huffman code algorithm

According to the article “Huffman Coding | Greedy Algo-3[1]”, we understand that the Huffman coding algorithm is calculated based on the frequency of occurrence of each character. Originally, we build a binary tree to demonstrate the result, so each node can only have two leaves. Now, in order to use digits to instead of bits, we need to image each node has ten leaves. Then, use the same method as in the article to build a tree. Using 26 characters as an example,

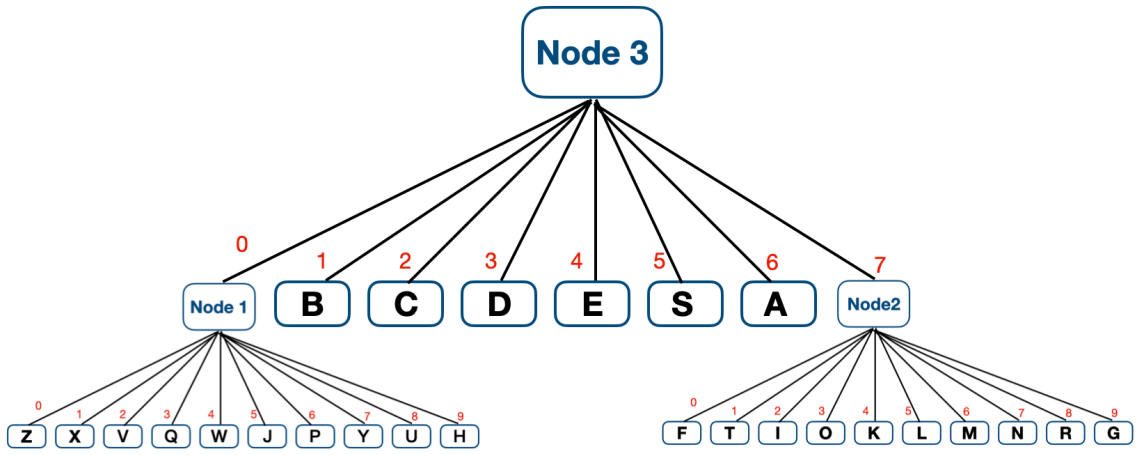
First of all, collect ten characters which has less frequencies and make a node to connect them.



Second, we make the same branch with node 2 and 10 characters.



Finally, we add the last 6 characters in this tree.



As the result, we can use less than two digits to represent a character.

Now, let us test an example to show the algorithm works.

“APPLE” → 60606754

When using the digits to get the string back, there is one thing need we need to remember: keep read the input until arrive a leave. Using another example,

Getting digits : 096060607, when running the first digit 0, we cannot stop by “Node 1” because it has leaves, we need to run the second digit 9 to get ‘H’. Finally, we will get the string “HAPPY”.

3. Whether a cryptographic protocol has a flaw or not

   According to the encryption protocol we know, the key is time-sensitive during the communication between the two communication devices. If you want to test each number and find the keys in a limited time, you need a super-efficient computer to do it. Therefore, it does not make sense to spend time and money to buy a super-efficient computer for the purpose of understanding non-confidential information. From the article “the many, many ways that cryptographic software can fail[2]”, we can see that in addition to the password agreement, there are four reasons that can invalidate the password software, such as bugs in crypto libraries, operating systems and apps, bad design, and misconfigurations or insecure default configurations.

Reference

[1] Huffman Coding: Greedy Algo-3. (2020, April 17). Retrieved from https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/

[2] Nabeel Y., (2017, January 18). The many, many ways that cryptographic software can fail. Retrieved from https://www.freecodecamp.org/news/why-does-cryptographic-software-fail-often-d660d3cdfdc5/