Henrique Tsuyoshi Yara
11796083

Lista 1

Exercício 1

1.1

Implementar em linguagem Python o algoritmo de exponenciação modular descrito abaixo

Resolução.
Está dentro do arquivo compactado.

1.2

Listar o seu número USP de 6 dígitos.

Resolução.
117960

1.3

Calcular e listar seu NUSP elevado a 2345, $\mod 6789$. E.g., para esse NUSP: $298765^{2345} \mod 6789 = 52326$ Sugestão: fazer testes com números de 2 ou 3 dígitos

Resolução.
$117960^{2345} \mod 6789 = 1761$

1.4

Justificar a complexidade de tempo de execução deste algoritmo que é O(log e).

Resolução.
Considerando o números de atribuições e comparações. Sabendo que $t = \lfloor log_2(e) \rfloor + 1$ é o número de bits do número e.

Na linha 1:
$\theta(1)$

No laço de t até 0:
$\theta(\lfloor log_2(e) \rfloor + 1) \times (\theta(1) + O(1))$

Na última linha:
$\theta(1)$

Temos a equação:
$f(e) = \theta(2) + O((1 + log_2(e)) \times 2)$
$f(e) = \theta(2) + O(2 + log_2(e) \times 2)$
$f(e) = O(4 + log_2(e) \times 2)$

Desconsiderando os valores constantes temos o tempo de complexidade:
$O(log_2(e))$

□

Exercício 2

2.1

Com parâmetro $5 \leq w \leq 10$, calcular um número inteiro primo absoluto maior que o inteiro correspondente aos 8 dígitos da sua data de nascimento *ddmmaaaa*. E.g., para a data 22/10/8899, os dígitos são 22108899. E 22109777, 221101061 são primos absolutos. Sugestão: fazer testes com números de 2 ou 3 dígitos.

Resolução.
Minha data: 29072001
O número inteiro primo absoluto é 29072009

2.2

Deduzir e justificar o tempo de execução deste algoritmo como função de w e de y.

Resolução.
Considerando o número de atribuições e comparações.
Sendo que a função $g(n-1)$ calcula t e c, e o tempo de complexidade é o número de zeros entre o bit menos significate (inclusivo) até o primeiro bit 1 (exclusivo) $g(n-1) = O(\lfloor \log_2 \lceil n-1 \rceil \rfloor)$:

Para obter t e c:
$g(y-1)$

No loop de 1 até w:
Escolher um testemunho:
$\theta(1)$
Atribuir valor à r_0 e r_1:
$\theta(2)$
No loop de 1 até t:
Comparar valores de r_j e r_{j-1}:
$O(3)$

Atribuir valor de $r_j + 1$:
$\theta(1)$

Temos a equação:
$f(y, w) = g(y-1) + O(w) \times (\theta(1) + \theta(2) + O(\lfloor \log_2 [y-1] \rfloor) \times (O(3) + \theta(1)))$
$f(y, w) = O(\lfloor \log_2 [y-1] \rfloor) + O(w) \times (\theta(3) + O(\log_2 [y-1]) \times O(4))$
$f(y, w) = O(\log_2 [y-1]) + O(3w) + O(4w) \times O(\log_2 [y-1])$

Desconsiderando os valores constantes temos o tempo de complexidade:
$f(y, w) = O(w)O(\log_2 [y-1]) + O(\log_2 [y-1]) + O(w)$

Exercício 3

3.1

Calcular e listar dois inteiros primos absolutos q, r, não necessariamente consecutivos, cada um maior que o inteiro correspondente aos 8 dígitos da sua data de nascimento ddmmaaaa. E.g., para a data 22/10/8899, os dígitos são 22108899.

Resolução.
$q = 29072881; r = 29072921$

3.2

Calcular e listar $n = q \times r$.
E.g., $22109776 \times 221101061 = 4888495153173397$

Resolução.
$n = q \times r = 29072921 \times 29072881 = 845233572555401$

3.3

Calcular e listar $\phi(n) = (q-1)(r-1)$(Função de Euler). E.g., $22109776 \times 221101060 = 4888494909962560$

Resolução.
$\phi = (q-1)(r-1) = 29072920 \times 29072880 = 845233514409600$

3.4

Sortear aleatoriamente e listar a chave secreta sdo RSA com pelo menos 10 dígitos. E.g., $s = 1234567899$

Resolução.
$s = 7290335707$

3.5

Calcular e listar a chave pública pcorrespondente a $s : p \times s = 1 \mod (n)$.
E.g., $1234567899^{-1} \mod 4888494909962560 = 3858608707133139$

Resolução.
$p = 310717853942428, s = 7290335707, 310717853942428 \times 7290335707 \mod 845233514409600 = 1$

3.6

Calcular e listar $p \times s \mod \phi(n) = ps$.
E.g., $1234567899 \times 3858608707133139 \mod 4888494909962560 = 1$

Resolução.
$p = 81213405628243, s = 7290335707, 81213405628243 \times 7290335707 \mod 845233514409600 = 1$

3.7

Seja x_0 o seu NUSP repetido várias vezes para completar pelo menos 15 dígitos.
E.g., para o NUSP 298765, os 15 dígitos são $298765298765298 = x_0$

Resolução.
$x_0 = 117960117960117$

3.8

Listar $x_0 = x_0$

Resolução.
$x_0 = 117960117960117$

3.9

Calcular e listar $y_0 = RSA(x_0, p) = (x_0)^p \mod n$.
E.g., $298765298765298^{3858608707133139} \mod 4888495153173397 = 180585179472907$

Resolução.
$y_0 = 55044909303457$

3.10

Calcular e listar $x_1 = RSA^{-1}(y_0)$.
E.g., $180585179472907^{1234567899} \mod 4888495153173397 = 298765298765298$

Resolução.
$x_1 = 117960117960117$

3.11

Calcular e listar $RSA(x_1)$.
E.g., $298765298765298^{3858608707133139} \mod 4888495153173397 = 180585179472907$

Resolução.
$y_1 = 55044909303457$