ĐỒ ÁN 1 - RSA

1 Kiểm tra số nguyên tố (4 điểm)

1.1 Định nghĩa số nguyên tố

Cho số nguyên dương $n \in \mathbb{Z}^+$. Khi đó, n được gọi là số nguyên tố khi và chỉ khi n có đúng 2 ước nguyên dương là 1 và chính nó.

Ví dụ:

- 2, 3, 5, 7 là số nguyên tố vì chúng có đúng 2 ước nguyên dương là 1 và chính nó.
- 1 không phải là số nguyên tố vì 1 chỉ có đúng 1 ước nguyên dương là 1.
- 4, 6, 8 không phải là số nguyên tố vì chúng có ít nhất 3 ước nguyên dương là 1, 2 và chính nó.

1.2 Gợi ý các thuật toán kiểm tra số nguyên tố

Sau đây là một số thuật toán kiểm tra số nguyên tố mà các bạn có thể thực hiện:

1. Thuật toán vét cạn.
2. Thuật toán Fermat.
3. Thuật toán Miller-Rabin.
4. Thuật toán AKS.

Các bạn vẫn được quyền sử dụng các thuật toán kiểm tra số nguyên tố khác không được nêu ở trên.

1.3 Yêu cầu bài toán

Trong phần này, sinh viên được yêu cầu thiết kế một chương trình dùng để kiểm tra một số nguyên dương $n \in \mathbb{Z}^+$ có phải là số nguyên tố hay không. Chương trình phải đảm bảo các yêu cầu sau:

- Sử dụng ngôn ngữ lập trình C++. Mọi ngôn ngữ lập trình khác nếu sinh viên sử dụng đều không được chấp nhận và sẽ nhận điểm 0 trong bài tập này.
- Chỉ được sử dụng các thư viện tiêu chuẩn có sẵn từ phiên bản C++17 trở xuống. Mọi thư viện khác nếu sinh viên cố ý sử dụng trong bài làm đều sẽ nhận điểm 0 trong bài tập này.
- Toàn bộ mã nguồn cho chương trình phải nằm trong một file duy nhất, đặt tên là `bai1.cpp`.
- Sau khi biên dịch mã nguồn thành file `a.exe`, chương trình phải được chạy thông qua lệnh sau:

  ```
  $ .\a.exe test.inp test.out
  ```

  Với `test.inp` là file chứa dữ liệu đầu vào, và `test.out` là file chứa kết quả đầu ra của chương trình.
- Thời gian chạy tối đa cho mỗi test case là **60 giây**. Sau thời gian này, bài làm của bạn sẽ bị dừng và test case đó không được tính điểm.

1.3.1 Các file chương trình

- File `test.inp` chứa duy nhất một số nguyên dương $n \in \mathbb{Z}^+$ được lưu bằng các chữ số thập lục phân in hoa dưới dạng big endian.
- Kết quả sau khi chạy chương trình sẽ được lưu trong file `test.out`. File này chứa số 0 nếu số ta kiểm tra không phải là số nguyên tố, và ngược lại chứa số 1 nếu số đó là số nguyên tố.

1.3.2 Ví dụ

test.inp	test.out
65	1

Giải thích: `0x65 = 101` là số nguyên tố nên kết quả trả ra là `1`.

test.inp	test.out
2406	0

Giải thích: `0x2406 = 9222 = 2 * 3 * 29 * 53` không phải là số nguyên tố nên kết quả trả ra là `0`.

1.3.3 Ràng buộc

- 35% số test ứng với 35% số điểm thoả mãn độ dài của số nguyên dương đầu vào là 64 bit.
- 25% số test ứng với 25% số điểm thoả mãn độ dài của số nguyên dương đầu vào là 128 bit.
- 20% số test ứng với 20% số điểm thoả mãn độ dài của số nguyên dương đầu vào là 256 bit.
- 15% số test ứng với 15% số điểm thoả mãn độ dài của số nguyên dương đầu vào là 512 bit.
- 5% số test ứng với 5% số điểm thoả mãn độ dài của số nguyên dương đầu vào là 1024 bit.

2 Sinh khoá cho hệ mã RSA (2.5 điểm)

2.1 Định nghĩa thuật toán sinh khoá của hệ mã RSA

Cho p và q là hai số nguyên tố lớn với $p \neq q$. Khi đó, $N = p \cdot q$, $\varphi(N) = (p-1) \cdot (q-1)$, và tồn tại cặp số nguyên dương (e, d) với $e, d < \varphi(N) < N$ thỏa $e \cdot d \equiv 1 \pmod{\varphi(N)}$. Từ đó, ta định nghĩa khoá công khai của hệ mã RSA là cặp số nguyên (N, e) và khoá bí mật là số nguyên d.

2.2 Yêu cầu bài toán

Trong phần này, sinh viên được yêu cầu thiết kế một chương trình dùng để sinh khoá bí mật d từ hai số nguyên tố lớn p, q và khoá công khai e cho trước. Chương trình phải đảm bảo các yêu cầu sau:

- Sử dụng ngôn ngữ lập trình C++. Mọi ngôn ngữ lập trình khác nếu sinh viên sử dụng đều không được chấp nhận và sẽ nhận điểm 0 trong bài tập này.
- Chỉ được sử dụng các thư viện tiêu chuẩn có sẵn từ phiên bản C++17 trở xuống. Mọi thư viện khác nếu sinh viên cố ý sử dụng trong bài làm đều sẽ nhận điểm 0 trong bài tập này.
- Toàn bộ mã nguồn cho chương trình phải nằm trong một file duy nhất, đặt tên là `bai2.cpp`.
- Sau khi biên dịch mã nguồn thành file `a.exe`, chương trình phải được chạy thông qua lệnh sau:

```
$ .\a.exe test.inp test.out
```

  Với `test.inp` là file chứa dữ liệu đầu vào, và `test.out` là file chứa kết quả đầu ra của chương trình.

- Thời gian chạy tối đa cho mỗi test case là **20 giây**. Sau thời gian này, bài làm của bạn sẽ bị dừng và test case đó không được tính điểm.

2.2.1 Các file chương trình

- File `test.inp` bao gồm 3 dòng. Dòng thứ nhất chứa số nguyên tố p. Dòng thứ hai chứa số nguyên tố q. Và dòng thứ ba chứa khoá công khai e. Ba con số mô tả trong file này được lưu bằng các chữ số thập lục phân in hoa dưới dạng big endian.
- Kết quả sau khi chạy chương trình được lưu trong file `test.out`. File này chứa số nguyên dương d nhỏ nhất được lưu bằng các chữ số thập lục phân in hoa dưới dạng big endian sao cho (pq, e) và d là một cặp khoá hợp lệ của hệ mã RSA. Tuy vậy, nếu ta không thể sinh được khoá bí mật như vậy, chương trình trả ra -1.

2.2.2 Ví dụ

test.inp	test.out
9D C1 17	60A7

Giải thích: Do `p = 0x9D = 157`, `q = 0xC1 = 193` nên $\varphi(N) = (p-1) \cdot (q-1) = 29952$. Vậy với `e = 0x17 = 23` thì ta tìm được `d = 24743 = 0x60A7` là số nguyên dương nhỏ nhất thoả yêu cầu bài toán.

test.inp	test.out
1A7B 1E2F 208	-1

Giải thích: Với `p = 0x1A7B = 6779`, `q = 0x1E2F = 7727` thì $\varphi(N) = (p-1) \cdot (q-1) = 52366828$. Mặc dù vậy, ước chung nhỏ nhất của e và $\varphi(N)$ là $gcd(e, \varphi(N)) = 4 \neq 1$ nên không tồn tại số nguyên dương d thoả yêu cầu. Do đó, chương trình trả ra -1.

2.2.3 Ràng buộc

- 40% số test ứng với 40% số điểm thoả mãn độ dài của số nguyên tố p đầu vào (hoặc q) là 64 bit.
- 30% số test ứng với 30% số điểm thoả mãn độ dài của số nguyên tố p đầu vào (hoặc q) là 128 bit.
- 15% số test ứng với 15% số điểm thoả mãn độ dài của số nguyên tố p đầu vào (hoặc q) là 256 bit.
- 10% số test ứng với 10% số điểm thoả mãn độ dài của số nguyên tố p đầu vào (hoặc q) là 512 bit.
- 5% số test ứng với 40% số điểm thoả mãn độ dài của số nguyên tố p đầu vào (hoặc q) là 1024 bit.
- Tất cả các test đều đảm bảo khoá công khai $e < \min(p, q)$.

3 Mã hoá và Giải mã cho hệ mã RSA (3.5 điểm)

3.1 Thuật toán mã hoá và giải mã của hệ mã RSA

Nhìn chung, hai thuật toán hoạt động khá giống nhau, nghĩa là ta lấy tin nhắn m (hay bản mã c) rồi đem mũ cho khoá công khai e (hay khoá bí mật d) trong modulo N.

- Thuật toán mã hoá: Cho cặp khoá công khai (N, e) được khởi tạo ở thuật toán sinh khoá, và tin nhắn $m \in \mathbb{Z}^+$ thỏa $m < N$. Khi đó, với tin nhắn m thì ta tính bản mã $c \in \mathbb{Z}^+$ thỏa $c < N$ và $c = m^e \pmod N$.
- Thuật toán giải mã: Cho một phần khoá công khai N và khoá bí mật d được khởi tạo ở thuật toán sinh khoá, và bản mã $c \in \mathbb{Z}^+$ thỏa $c < N$. Khi đó, với bản mã c thì ta tính được tin nhắn ban đầu $m \in \mathbb{Z}^+$ thỏa $m < N$ và $m = c^d \pmod N$.

3.2 Yêu cầu bài toán

Do thuật toán mã hoá và giải mã của RSA khá tường minh nên sinh viên sẽ làm một bài toán khác thử thách hơn. Trong phần này, sinh viên được yêu cầu thiết kế một chương trình dùng để tìm bản mã c_j trong một mảng các bản mã cho trước khi thực hiện mã hoá tin nhắn m_i sử dụng khoá công khai (n, e) cho trước. Chương trình phải đảm bảo các yêu cầu sau:

- Sử dụng ngôn ngữ lập trình C++. Mọi ngôn ngữ lập trình khác nếu sinh viên sử dụng đều không được chấp nhận và sẽ nhận điểm 0 trong bài tập này.
- Chỉ được sử dụng các thư viện tiêu chuẩn có sẵn từ phiên bản C++17 trở xuống. Mọi thư viện khác nếu sinh viên cố ý sử dụng trong bài làm đều sẽ nhận điểm 0 trong bài tập này.
- Toàn bộ mã nguồn cho chương trình phải nằm trong một file duy nhất, đặt tên là `bai3.cpp`.
- Sau khi biên dịch mã nguồn thành file `a.exe`, chương trình phải được chạy thông qua lệnh sau:

```
$ .\a.exe test.inp test.out
```

  Với `test.inp` là file chứa dữ liệu đầu vào, và `test.out` là file chứa kết quả đầu ra của chương trình.
- Thời gian chạy tối đa cho mỗi test case là **60 giây**. Sau thời gian này, bài làm của bạn sẽ bị dừng và test case đó không được tính điểm.

3.2.1 Các file chương trình

- File `test.inp` bao gồm các dòng như sau:
  - Dòng thứ nhất chứa 2 số nguyên dương x và y được lưu bằng các chữ số thập phân dưới dạng big endian, lần lượt là số lượng tin nhắn và số lượng bản mã có sẵn.
  - Dòng thứ hai chứa 2 số nguyên N và e được lưu bằng các chữ số thập lục phân in hoa dưới dạng big endian, là khoá công khai của hệ mã RSA.

  - Dòng thứ i trong số x dòng tiếp theo chứa tin nhắn m_i được lưu bằng các chữ số thập lục phân in hoa dưới dạng big endian.
  - Dòng thứ j trong số y dòng tiếp theo chứa bản mã c_j được lưu bằng các chữ số thập lục phân in hoa dưới dạng big endian.

- Kết quả sau khi chạy chương trình được lưu trong file `test.out`. File này bao gồm 1 dòng chứa x phần tử, mỗi phần tử lưu vị trí j trong mảng bản mã cho trước thoả $c_j = m_i^e \pmod N$. Nếu không có vị trí thoả yêu cầu trên, phần tử đó được lưu bằng giá trị -1.

3.2.2 Ví dụ

test.inp	test.out
3 10 105B 1F 42 A8 5D A5F E1C EF3 67A 357 92B E46 84A 7A1 C50	-1 4 9

Giải thích: Có `m = [0x42, 0xA8, 0x5D] = [66, 168, 93]` và `c = [2655, 3612, 3827, 1658, 855, 2347, 3654, 2122, 1953, 3152]`. Ngoài ra, có khoá công khai $(n, e) =$ (`0x105B, 0x1F`) $= (4187, 31)$.

- Với $m_0 = 66$, không tồn tại c_j trong tập bản mã cho trước thoả yêu cầu đề bài nên in ra -1.
- Với $m_1 = 168$ thì $c_4 = 855 = 168^{31} \pmod{4187} = m_1^e \pmod N$.
- Với $m_2 = 93$ thì $c_9 = 3152 = 93^{31} \pmod{4187} = m_2^e \pmod N$.

3.2.3 Ràng buộc

- $2 \le x \le 5$, $10 \le y \le 20$.
- 30% số test ứng với 30% số điểm thoả mãn độ dài của n tính theo bit nhỏ hơn hoặc bằng 64 bit, kí hiệu $|n| \le 64$.
- 30% số test ứng với 30% số điểm thoả mãn $64 < |n| \le 128$.
- 20% số test ứng với 20% số điểm thoả mãn $128 < |n| \le 256$.
- 10% số test ứng với 10% số điểm thoả mãn $256 < |n| \le 512$.
- 10% số test ứng với 10% số điểm thoả mãn $512 < |n| \le 1024$.

4 Các quy định khác về đồ án

- Đồ án này được thực hiện cá nhân. Nếu phát hiện bất kì hành vi sao chép bài nào của các bạn cùng môn học, toàn bộ phần điểm thực hành của những sinh viên có liên quan sẽ được đưa về 0.
- Thời gian thực hiện là 3 tuần tính từ lúc đồ án được chính thức đăng lên trên hệ thống quản lý môn học Moodle.
- Cấu trúc file để nộp đồ án như sau:

```
MSSV.zip
└── MSSV-Project1
    ├── bai1.cpp
    ├── bai2.cpp
    └── bai3.cpp
```

  Trong đó:

  - Thay cụm `MSSV` thành mã số sinh viên của người nộp. Ví dụ, `22127001.zip` hay `22127001-Project1`.
  - `MSSV-Project1` là thư mục chứa các file mã nguồn trong đồ án này.
  - `.zip` là định dạng nén cho bài làm.
  - `bai1.cpp`, `bai2.cpp`, `bai3.cpp` là các file chứa toàn bộ mã nguồn ứng với từng yêu cầu bài tập nêu trên.

- Nộp file `MSSV.zip`. Nếu sinh viên nộp sai quy định thì toàn bộ phần đồ án này sẽ bị điểm 0.
- Nếu chương trình không biên dịch được thì sẽ bị điểm 0 ở bài tập đó.
- Nếu chương trình biên dịch được nhưng không trả ra được kết quả (do gặp lỗi khi chạy hoặc quá thời gian quy định) ở test nào thì test đó không được tính điểm.
- Thư viện chuẩn của `C++` (`C++` Standard Library) là các thư viện được liệt kê ở `https://en.cppreference.com/w/cpp/standard_library`. Sinh viên cần lưu ý về phiên bản C++ để lựa chọn thư viện thích hợp.
- Mọi thắc mắc về đồ án này, vui lòng gửi qua email: `nvqhuy@fit.hcmus.edu.vn`.