**TÀI LIỆU CHUYÊN TIN**

**CHUYÊN ĐỀ 1. THUẬT TOÁN VÀ PHÂN TÍCH THUẬT TOÁN**

1. Thuật toán

2. Phân tích thuật toán

**CHUYÊN ĐỀ 2. KIẾN THỨC NỀN TẢNG SỐ HỌC VÀ TỔ HỢP**

*Chuyên đề này sẽ cung cấp các kiến thức toán học cơ bản nhưng thiết yếu, làm nền tảng vững chắc cho việc tiếp cận các thuật toán và giải quyết các bài toán tin học.*

**1. Số nguyên tố**

**1.1. Khái niệm**

Số nguyên tố là số tự nhiên lớn hơn 1, chỉ chia hết cho 1 và chính nó.

*Ví dụ: 2, 3, 5, 7 là các số nguyên tố; 4 không phải là số nguyên tố vì chia hết cho 2; Số 2 là số nguyên tố nhỏ nhất và là số nguyên tố chẵn duy nhất.*

*10?*

*1 2 3 4 5*

**1.2. Cách kiểm tra số nguyên tố**

- Duyệt từ 2 đến phần nguyên căn bậc hai của n, nếu tồn tại số mà n chia hết thì n không phải là nguyên tố.

- Thuật toán kiểm tra số nguyên tố:

*Bước 1: Nếu n < 2 → kết luận n không phải là số nguyên tố (False).*

*Bước 2: Nếu n < 4 → kết luận n là số nguyên tố (True).*

*Bước 3: Duyệt i từ 2 đến (n-1):*

*Nếu n % i == 0 → kết luận n không phải là số nguyên tố (False).*

*Bước 4: Kết luận n là số nguyên tố (True).*

**1.3. Bài tập thực hành**

- Kiểm tra số n có phải là số nguyên tố hay không.

Ý tưởng: Duyệt i từ 2 đến int(n\*\*0.5), nếu n không chia hết cho bất kì số i nào thì n là số nguyên tố.

|  |  |
| --- | --- |
| **Input** | **Output** |
| Nhập n = 10 | False |
| Nhập n = 13 | True |

- Tính tổng và in tất cả số nguyên tố từ 1 đến n.

*Ý tưởng: Duyệt i từ 1 đến n, nếu i là số nguyên tố thì in.*

|  |  |
| --- | --- |
| **Input** | **Output** |
| Nhập n = 10 | Các số nguyên tố: 2 3 5 7  Tổng các số nguyên tố: 17 |
| Nhập n = 20 | Các số nguyên tố: 2 3 5 7 11 13 17 19  Tổng các số nguyên tố: 77 |

- Đếm số lượng số nguyên tố trong đoạn [a, b].

Ý tưởng: Duyệt i từ a đến b, kiểm tra nếu i là số nguyên tố thì tăng đếm 1 đơn vị.

5 6 7 8 9 10 11 12 13 14 15

Dem = 0 +1+1+1+1

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Giải thích** |
| Nhập a = 5  Nhập b = 15 | 4 | *Từ 5 đến 10 có 4 số nguyên tố: 5 7 11 13* |
| Nhập a = 10  Nhập b = 30 | 6 | *Từ 10 đến 30 có 6 số nguyên tố: 11 13 17 19 23 29* |

- Tìm số nguyên tố lớn nhất nhỏ hơn n.

Ý tưởng:

Cách 1: Duyệt từ 2 đến (n-1) để tìm số nguyên tố gần n nhất.

Cách 2: Duyệt ngược từ (n-1) về 2 để tìm số nguyên tố gần n nhất.

|  |  |
| --- | --- |
| **Input** | **Output** |
| Nhập n = 10 | 7 |
| Nhập n = 20 | 19 |
| Nhập n = 2 | -1 |

- Tìm số nguyên tố thứ k.

Ý tưởng: Duyệt từng số, đếm số nguyên tố gặp được.

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Giải thích** |
| Nhập k = 5 | 11 | 2 3 5 7 **11** |
| Nhập k = 8 | 19 | 2 3 5 7 11 13 17 **19** |

- Phân tích thừa số nguyên tố. *(Một số nguyên có thể được biểu diễn dưới dạng tích của các số nguyên tố. Ví dụ: 60 = 2 × 2 × 3 × 5).*

Ý tưởng: Dùng phép chia liên tục với các ước nguyên tố.

- Số siêu nguyên tố theo chữ số.

*Viết chương trình kiểm tra một số nguyên dương có phải là số siêu nguyên tố hay không. Một số được gọi là số siêu nguyên tố nếu:*

*+ Bản thân nó là số nguyên tố.*

*+ Mỗi chữ số của nó cũng là số nguyên tố.*

*Ví dụ:*

*Nhập 23 → Là số siêu nguyên tố.*

*Nhập 29 → Không phải số siêu nguyên tố (vì 9 không nguyên tố).*

- Số siêu nguyên tố theo từng phần đầu.

*Viết chương trình kiểm tra một số nguyên dương n ≥ 2 có phải là số siêu nguyên tố hay không. Một số được gọi là số siêu nguyên tố nếu:*

*+ Bản thân nó là số nguyên tố.*

*+ Tất cả các phần bên trái liên tiếp của nó (khi viết theo thứ tự từ trái sang phải) cũng là số nguyên tố.*

*Nhập: 233 (2 → nguyên tố, 23 → nguyên tố, 233 → nguyên tố). Kết luận: 233 là số siêu nguyên tố.*

*Nhập: 221 (2 → nguyên tố, 22 → không nguyên tố). Kết luận: 221 là không số siêu nguyên tố.*

- Tìm tất cả cặp số nguyên tố sinh đôi trong đoạn [a, b]. (hai số nguyên tố cách nhau 2 đơn vị. Ví dụ: (3, 5), (11, 13).

Ý tưởng: Duyệt từ a đến b-2, kiểm tra (i, i+2) đều nguyên tố.

- Tìm số nguyên tố có tổng chữ số là số nguyên tố.

**2. Số đặc biệt**

**2.1. Các khái niệm**

**Số hoàn hảo:** Là số tự nhiên mà tổng các ước số nguyên dương thật của nó (tức là không kể chính nó) bằng chính nó (ví dụ: 6=1+2+3).

**Số chính phương:** Là bình phương của một số nguyên (ví dụ: 4,9,16). Cách kiểm tra bằng cách lấy căn bậc hai và kiểm tra xem kết quả có phải là số nguyên không.

**Số Armstrong (Số narcissistic):** Là số có N chữ số mà tổng lũy thừa N của mỗi chữ số bằng chính nó (ví dụ: 153=13+53+33).

**Số Palindrome:** Là số mà đọc xuôi hay đọc ngược đều giống nhau (ví dụ: 121,12321). Có thể kiểm tra bằng cách chuyển số sang chuỗi và so sánh với chuỗi đảo ngược.

**Số tam giác:** Là tổng của các số tự nhiên liên tiếp từ 1 đến n.

Công thức: Tn = n×(n+1)/2.

Ví dụ: Với n = 4, dãy số tam giác là: 1, 3, 6, 10.

|  |  |  |
| --- | --- | --- |
| **n** | **Công thức** | **Giá trị** |
| 1 | 1 | 1 |
| 2 | 1+2 | 3 |
| 3 | 1+2+3 | 6 |
| 4 | 1+2+3+4 | 10 |

**2.2. Bài tập thực hành**

- Kiểm tra một số có phải là số hoàn hảo/chính phương/Armstrong/Palindrome/tam giác không.

- Tìm/tính tổng/đếm các số đặc biệt trong đoạn [a, b].

**3. Ước số, bội số**

**3.1. Khái niệm**

**- Ước số:** Số a là ước của b nếu b chia hết cho a.

**- Bội số:** Số b là bội của a nếu b chia hết cho a.

**- Ước chung lớn nhất (UCLN):** Số lớn nhất chia hết cho cả hai (hoặc nhiều) số.

**- Bội chung nhỏ nhất (BCNN):** Số nhỏ nhất là bội của cả hai (hoặc nhiều) số.

**3.2. Cách tính**

**- Thuật toán Euclid:** Phương pháp hiệu quả để tìm GCD của hai số.

**- Công thức liên hệ GCD và LCM:** LCM(a,b)=(a×b)/GCD(a,b).

**3.3. Bài tập thực hành**

- Liệt kê tất cả các ước số của một số N.

- Đếm số lượng ước số của N.

- Tính tổng các ước số của N.

- Tìm GCD và LCM của hai số A và B.

- Tìm số có đúng k ước số

- Tìm số có tổng các ước là số nguyên tố

- Bài toán nâng cao: Tìm số có nhiều ước nhất trong một khoảng [1,N].

**4. Dãy số Fibonacci**

**4.1. Khái niệm**

Dãy Fibonacci được định nghĩa bởi công thức truy hồi: F(0)=0, F(1)=1, và F(n)=F(n−1)+F(n−2) với n≥2.

**4.2. Cách tính**

**- Đệ quy:** Cài đặt trực tiếp theo công thức (cần lưu ý về hiệu suất do tính toán trùng lặp).

**- Quy hoạch động (lặp):** Sử dụng mảng hoặc biến để lưu trữ các giá trị đã tính, tránh tính toán lại. Đây là phương pháp hiệu quả hơn.

**4.3. Bài tập thực hành**

- Tìm số Fibonacci thứ N.

- Kiểm tra một số có phải là số Fibonacci không.

- Tìm số Fibonacci nhỏ nhất lớn hơn N.

- Tính tổng các số Fibonacci nhỏ hơn n.

- Tổng n số Fibonacci đầu tiên.

- Tìm số Fibonacci chẵn trong khoảng

**5. Lý thuyết tập hợp**

**5.1. Khái niệm**

**- Tập hợp:** Một bộ sưu tập các đối tượng riêng biệt.

**- Phần tử:** Các đối tượng trong tập hợp.

**- Tập con:** Mọi phần tử của tập này cũng là phần tử của tập kia.

**5.2. Các phép toán:**

**- Hợp (∪):** Tập hợp chứa tất cả các phần tử của cả hai tập.

**- Giao (∩):** Tập hợp chứa các phần tử chung của cả hai tập.

**- Hiệu (∖):** Tập hợp chứa các phần tử chỉ có trong tập thứ nhất mà không có trong tập thứ hai.

**- Phần bù:** Các phần tử không thuộc tập hợp đã cho trong không gian mẫu.

**5.3. Ứng dụng trong lập trình (sử dụng kiểu dữ liệu set trong Python)**

- Khởi tạo tập hợp.

- Thêm/xóa phần tử.

- Kiểm tra phần tử thuộc tập hợp (độ phức tạp O(1) trung bình).

- Thực hiện các phép toán hợp, giao, hiệu trực tiếp.

**5.3. Bài tập thực hành**

- Tính hợp, giao, hiệu của hai tập hợp.

- Đếm số phần tử trong tập hợp sau khi thực hiện các phép toán.

- Kiểm tra phần tử thuộc tập hợp.

- Liệt kê tất cả tập con của tập hợp.

- Tìm tập con có tổng bằng S.

**6. Hệ đếm**

**6.1. Khái niệm**

**- Hệ đếm cơ số bất kỳ:** Hệ thập phân (cơ số 10), hệ nhị phân (cơ số 2), hệ bát phân (cơ số 8), hệ thập lục phân (cơ số 16).

**- Chuyển đổi giữa các hệ đếm:** Đặc biệt quan trọng là chuyển đổi giữa hệ 10 và hệ 2.

**6.2. Bài tập thực hành**

- Chuyển một số từ hệ 10 sang hệ 2, 8, 16 và ngược lại.

- Cộng hai số trong hệ nhị phân.

- Kiểm tra một số nhị phân có phải là số đối xứng (palindrome) không.

- Tìm số palindrome trong hệ cơ số k.

- Tìm số lớn nhất có n chữ số trong hệ cơ số k.

- Đếm số bit 1 trong biểu diễn nhị phân.

- Tìm số nhị phân có k bit 1.

### 7. Tổ hợp và chỉnh hợp

**7.1. Các khái niệm**

**- Giai thừa (Factorial):** n!=1×2×⋯×n.

**- Hoán vị (Permutation):** Sắp xếp n phần tử khác nhau. Số hoán vị: P(n)=n!.

**- Chỉnh hợp (Arrangement):** Chọn k phần tử từ n và sắp xếp chúng. Số chỉnh hợp: A(n,k)=n!/(n−k)!.

**- Tổ hợp (Combination):** Chọn k phần tử từ n mà không quan tâm thứ tự. Số tổ hợp: C(n,k)=n!/(k!×(n−k)!).

**- Hệ số nhị thức (Binomial Coefficient):** Thường được ký hiệu là (kn​), chính là C(n,k).

**7.2. Các nguyên lý đếm cơ bản:**

**- Nguyên lý cộng:** Nếu có k cách thực hiện hành động A và m cách thực hiện hành động B (không trùng lặp), thì có k+m cách thực hiện một trong hai hành động.

**- Nguyên lý nhân:** Nếu có k cách thực hiện hành động A và m cách thực hiện hành động B (sau khi A đã được thực hiện), thì có k×m cách thực hiện cả hai hành động theo thứ tự.

**- Nguyên lý Dirichlet (Nguyên lý Chuồng bồ câu - Pigeonhole Principle):** Nếu nhốt N con chim vào M chuồng (N>M), thì ít nhất một chuồng có nhiều hơn một con chim. Ứng dụng trong các bài toán chứng minh sự tồn tại.

**7.3. Bài tập thực hành**

- Tính số cách chọn k phần tử từ n phần tử.

- Tính số cách sắp xếp n phần tử.

- Giải bài toán đếm số cách chọn đội.

- Tính C(n,k) với n,k lớn (sử dụng modulo).

- Bài toán chia kẹo (Stars and Bars).

- Số cách phân tích n thành tổng k số dương.

**CHUYÊN ĐỀ 3. SẮP XẾP**

1. Phát biểu bài toán

2. Các thuật toán sắp xếp thông dụng

**CHUYÊN ĐỀ 4. TÌM KIẾM**

1. Bài toán tìm kiếm

2. Các thuật toán tìm kiếm thông dụng

**CHUYÊN ĐỀ 5. ĐỆ QUY**

1. Khái niệm về đệ quy

2. Giải thuật về đệ quy

**CHUYÊN ĐỀ 6. THIẾT KẾ GIẢI THUẬT**

1. Quay lui

2. Nhánh và cận

3. Tham ăn

4. Chia để trị

5. Quy hoạch động

**CHUYÊN ĐỀ 7. CÁC THUẬT TOÁN TRÊN ĐỒ THỊ**

1. Các khái niệm cơ bản

2. Biểu diễn đồ thị

3. Các thuật toán tìm kiếm trên đồ thị

4. Tính liên thông của đồ thị

5. Vài ứng dụng của DFS và BFS

6. Đồ thị Euler và độ thị Hamilton