**BÀI TẬP THỰC HÀNH 1**

***Mô tả bài toán***

Luật Horner là một phương pháp đơn giản để tính giá trị của đa thức tại một điểm xác định. Phương pháp này được đặt tên theo nhà toán học người Anh William George Horner.

Luật Horner cho phép tính giá trị của đa thức bậc n bất kỳ tại một điểm xác định x0 với độ phức tạp O(n). Phương pháp này cũng được sử dụng để rút gọn đa thức bằng cách tìm thừa số chung lớn nhất của các hệ số.

***Dữ liệu đầu vào và dữ liệu đầu ra***

Dữ liệu đầu vào:

- A: một mảng chứa các hệ số của đa thức theo thứ tự từ bậc cao nhất đến bậc thấp nhất, A[0] là hệ số của bậc cao nhất, A[n] là hệ số của bậc thấp nhất

- x0: một số thực, điểm xác định để tính giá trị của đa thức

Dữ liệu đầu ra: Giá trị của đa thức tại điểm x0

**INPUT: 2 -3 4 -5**

**OUTPUT: 7**

**BÀI TẬP THỰC HÀNH 2**

***Mô tả bài toán***

Phép nhân ma trận là một phép tính cơ bản trong đại số tuyến tính. Giải thuật Strassen là một phương pháp hiệu quả để nhân hai ma trận vuông cùng bậc. Độ phức tạp thời gian của phương pháp Strassen là O(nlog27), nhanh hơn so với phương pháp thông thường O(n3).

***Dữ liệu đầu vào và dữ liệu đầu ra***

Dữ liệu đầu vào:

- A, B: hai ma trận vuông cùng bậc n

- n: kích thước của ma trận vuông

Dữ liệu đầu ra: C: ma trận tích của A và B

**BÀI TẬP THỰC HÀNH 3**

***Mô tả bài toán***

Phép nhân đa thức là một phép tính cơ bản trong toán học. Giải thuật Strassen là một phương pháp hiệu quả để nhân hai đa thức có bậc nhỏ hơn hoặc bằng 3. Độ phức tạp thời gian của phương pháp Strassen là O(nlog27), nhanh hơn so với phương pháp thông thường O(n2).

***Dữ liệu đầu vào và dữ liệu đầu ra***

Dữ liệu đầu vào:

- A, B: hai đa thức cùng bậc n hoặc bậc nhỏ hơn hoặc bằng 3.

- n: bậc của đa thức

Dữ liệu đầu ra: C: đa thức tích của A và B

**BÀI TẬP THỰC HÀNH 4**

***Mô tả bài toán***

Quicksort là một thuật toán sắp xếp nhanh và hiệu quả trong việc sắp xếp một mảng các phần tử. Thuật toán QuickSort được phát minh bởi Tony Hoare vào năm 1960. Quicksort là một phương pháp chia để trị đệ quy. Quicksort chọn một phần tử trong mảng làm pivot và phân chia các phần tử còn lại thành hai mảng: các phần tử bé hơn pivot và các phần tử lớn hơn pivot. Tiếp theo, thuật toán đệ quy được áp dụng đối với hai mảng này.

***Dữ liệu đầu vào và dữ liệu đầu ra***

Dữ liệu đầu vào:

- arr: một mảng các số nguyên

Dữ liệu đầu ra:

- arr: mảng đầu vào đã được sắp xếp tăng dần

**BÀI TẬP THỰC HÀNH 5**

***Mô tả bài toán***

Quicksort là một thuật toán sắp xếp nhanh và hiệu quả trong việc sắp xếp một mảng các phần tử. Thuật toán QuickSort được phát minh bởi Tony Hoare vào năm 1960. Quicksort là một phương pháp chia để trị đệ quy. Quicksort chọn một phần tử trong mảng làm pivot và phân chia các phần tử còn lại thành hai mảng: các phần tử bé hơn pivot và các phần tử lớn hơn pivot. Tiếp theo, thuật toán đệ quy được áp dụng đối với hai mảng này. Trong thuật toán quicksort, việc chọn pivot là rất quan trọng. Khi chọn pivot tốt, thuật toán sẽ đạt hiệu quả cao.

Trong bài toán này, chúng ta sẽ cài đặt thuật toán quicksort với việc chọn vị trí mốc ngẫu nhiên làm pivot.

***Dữ liệu đầu vào và dữ liệu đầu ra***

Dữ liệu đầu vào

- arr: một mảng các số nguyên

Dữ liệu đầu ra:

- arr: mảng đầu vào đã được sắp xếp tăng dần

**BÀI THỰC HÀNH SỐ 6**

***Mô tả bài toán***

Bài toán tám quân hậu là bài toán đặt tám quân hậu trên bàn cờ vua kích thước 8×8 sao cho không có quân hậu nào có thể "ăn" được quân hậu khác, hay nói khác đi không quân hậu nào có để di chuyển theo quy tắc cờ vua. Màu của các quân hậu không có ý nghĩa trong bài toán này. Như vậy, lời giải của bài toán là một cách xếp tám quân hậu trên bàn cờ sao cho không có hai quân nào đứng trên cùng hàng, hoặc cùng cột hoặc cùng đường chéo. Bài toán tám quân hậu có thể tổng quát hóa thành bài toán đặt n quân hậu trên bàn cờ n×n(n ≥ 1).

***Dữ liệu đầu vào và dữ liệu đầu ra***

Dữ liệu đầu vào là số n (số dòng và số cột của bàn cờ vua)

Dữ liệu ra là số cách xếp quân hậu vào bàn cờ.

Ví dụ:

**INPUT: 7**

**OUTPUT: 40**

**BÀI THỰC HÀNH SỐ 7**

***Mô tả bài toán***

Tháp Hà Nội là trò chơi dịch chuyển 1 tháp có x tầng từ cột này sang cột kia. có 3 cột,2 cột trống và 1 cột chứa tháp, tầng dưới lớn hơn tầng trên, không được đặt tầng lớn lên tầng nhỏ, mỗi bước di chuyển 1 tầng.

***Dữ liệu đầu vào và đầu ra***

Ví dụ:

**INPUT: 6**

**OUTPUT: 63**

Ví dụ:

**INPUT: 1**

**OUTPUT: 1**

**BÀI TẬP THỰC HÀNH 8**

***Mô tả bài toán***

Bài toán balo là một bài toán tối ưu hóa về cách sắp xếp các vật phẩm vào trong một balo với dung tích giới hạn nhất định. Bài toán được sử dụng rộng rãi trong lĩnh vực quản lý chuỗi cung ứng và vận tải.

Trong bài toán này, chúng ta sẽ giải quyết hai bài toán con của bài toán balo:

- Balo 1: Cho trước danh sách các vật phẩm có giá trị và trọng lượng khác nhau. Hãy chọn các vật phẩm để đặt vào một balo có dung tích giới hạn, sao cho giá trị của các vật phẩm được chọn là lớn nhất có thể.

- Balo 2: Tương tự như balo 1, nhưng mỗi vật phẩm có một số lượng giới hạn, tức là chỉ có thể chọn số lượng nhất định của vật phẩm đó.

***Dữ liệu đầu vào và dữ liệu đầu ra***

Dữ liệu đầu vào:

* **Balo 1 :**

**INPUT:**

3 10

3 5 7

* **Balo 2 :**

**INPUT:**

4 11

3 4

5 6

6 5

1 3

Dữ liệu đầu ra:

* **Balo 1**

**OUTPUT:**

10

0 2

* **Balo 2**

**OUTPUT:**

13

1 1 0 1

**BÀI TẬP THỰC HÀNH 10**

***Mô tả bài toán:***

Cho hai chuỗi X và Y, tìm chuỗi chung dài nhất (LCS - Longest Common Subsequence) của chúng.

Ví dụ:

- X = "ABCDEF"

- Y = "ACDFG"

- Kết quả: chuỗi chung dài nhất là "ACDF"

***Dữ liệu đầu vào và dữ liệu đầu ra***

Dữ liệu đầu vào: hai chuỗi X và Y.

Dữ liệu đầu ra: chuỗi chung dài nhất của X và Y.

**BÀI TẬP THỰC HÀNH 11**

***Mô tả bài toán:***

Bài toán xếp lịch (Scheduling) là bài toán tìm cách xếp các công việc (jobs) vào các thời điểm trong một khoảng thời gian nhất định sao cho thỏa mãn các ràng buộc về thời gian hoàn thành và tối ưu hóa mục tiêu nào đó, ví dụ như tối ưu số lượng công việc hoàn thành, tối ưu thời gian hoàn thành, hoặc tối ưu tổng giá trị của các công việc.

***Dữ liệu đầu vào và dữ liệu đầu ra***

Dữ liệu đầu vào:

- Một danh sách các công việc (jobs), mỗi công việc có thời gian thực hiện và thời gian kết thúc.

- Khoảng thời gian tổng thể (tổng thời gian) mà các công việc phải được hoàn thành.

Dữ liệu đầu ra:

- Một lịch trình thực hiện các công việc sao cho tối ưu hóa mục tiêu được đề ra.

**BÀI TẬP THỰC HÀNH 12&13**

***Mô tả bài toán:***

Bài toán tìm đường đi ngắn nhất là một trong những bài toán quan trọng trong lý thuyết đồ thị. Đây là bài toán về việc tìm đường đi từ một đỉnh xuất phát đến một đỉnh đích sao cho tổng trọng số của các cạnh trên đường đi là nhỏ nhất.

***Dữ liệu đầu vào và dữ liệu đầu ra:***

Dữ liệu đầu vào: đồ thị có trọng số, với số đỉnh và số cạnh được biểu diễn bằng các số nguyên dương. Các cạnh được biểu diễn bằng cặp (u, v, w) trong đó u và v là các đỉnh được nối với nhau bởi cạnh có trọng số w.

Dữ liệu đầu ra: đường đi ngắn nhất từ đỉnh xuất phát đến đỉnh đích, với tổng trọng số nhỏ nhất.

**BÀI TẬP THỰC HÀNH 14**

***Mô tả bài toán:***

Bài toán đổi tiền là một bài toán trong lĩnh vực tiền tệ và tài chính, đặc biệt là trong các hoạt động mua bán. Bài toán đặt ra là cho trước số tiền cần đổi và danh sách các loại tiền có sẵn, mục tiêu là tìm ra cách đổi ít tiền nhất để được số tiền cần đổi.

***Dữ liệu đầu vào và dữ liệu đầu ra:***

Dữ liệu đầu vào:

- Số tiền cần đổi (total\_amount) - số nguyên dương.

- Danh sách các loại tiền có sẵn (coins) - một mảng các số nguyên dương biểu thị giá trị của từng đồng tiền.

Dữ liệu đầu ra:

- Số lượng đồng tiền cần đổi của mỗi loại tiền (coin\_counts) - một mảng các số nguyên không âm biểu thị số lượng đồng tiền cần đổi của từng loại tiền.

- Tổng số đồng tiền cần đổi (total\_coin\_count) - một số nguyên không âm biểu thị tổng số đồng tiền cần đổi.