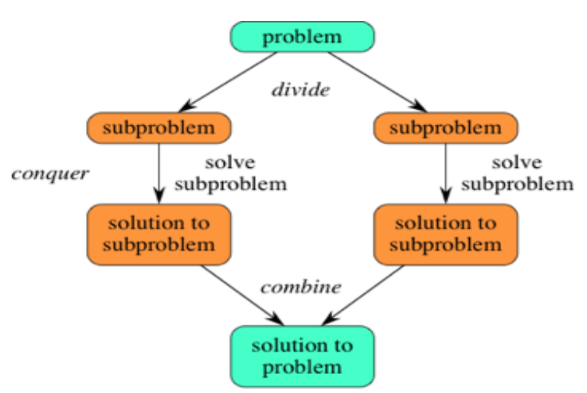
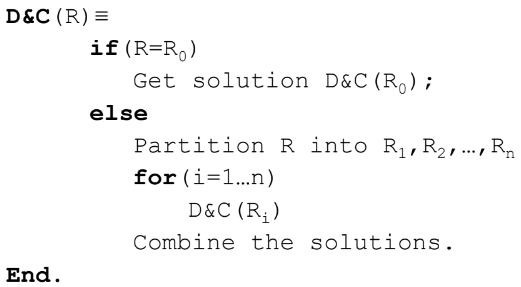
**§ Homework: Divide and Conquer §**

**Phần 1: Lý thuyết phương pháp**

Phương pháp chia để trị là một kỹ thuật giải thuật phổ biến trong lĩnh vực khoa học máy tính, được sử dụng để giải quyết các bài toán phức tạp bằng cách chia chúng thành các bài toán con nhỏ hơn, giải quyết từng bài toán con và kết hợp các kết quả con lại để tìm ra kết quả cuối cùng. Thường được sử dụng để giảm độ phức tạp tính toán của các bài toán lớn.

Lược đồ tổng quát:





*D&C(R)* là thuật toán của bài toán P với khoảng giá trị R.

Lược đồ tổng quát của phương pháp chia để trị bao gồm ba bước chính:

1. Chia bài toán thành các bài toán con nhỏ hơn, tương tự như bước chia bánh trong một bảnh mỳ.
2. Giải quyết các bài toán con này độc lập với nhau.
3. Kết hợp các kết quả con lại để tìm ra kết quả cuối cùng của bài toán ban đầu.

Ví dụ về phương pháp chia để trị là giải thuật Merge Sort, một giải thuật sắp xếp dãy số. Giải thuật này hoạt động theo các bước sau:

1. Chia dãy số ban đầu thành hai dãy số con bằng cách chia giữa dãy số.
2. Sắp xếp các dãy số con độc lập với nhau bằng cách sử dụng giải thuật Merge Sort đệ quy.
3. Trộn hai dãy số con đã sắp xếp để tạo ra dãy số hoàn chỉnh đã sắp xếp.

Đây là một ví dụ đơn giản về phương pháp chia để trị, nhưng nó rất hiệu quả và được sử dụng rộng rãi trong các giải thuật tính toán phức tạp khác.

Demo chương trình trong foder *Coding/Hw4/MergeSort.java*

**Phần 2: Bài tập tư duy**

**Exercise 1. Stock Pricing Problem:**

Bài toán yêu cầu tìm ngày mua và ngày bán cổ phiếu để đạt được lợi nhuận tối đa. Ta sử dụng phương pháp chia để trị để giải quyết bài toán trên: Input đầu vào sẽ là 1 arr chứa giá cổ phiếu của công ty qua n ngày liên tiếp

Ta sẽ chia mảng giá cổ phiếu thành hai nửa và tìm giá trị lớn nhất và nhỏ nhất trong mỗi nửa. Sau đó, chúng ta so sánh lợi nhuận của việc mua ở giá thấp nhất trong nửa đầu và bán ở giá cao nhất trong nửa sau, và lợi nhuận của việc mua ở giá thấp nhất trong nửa sau và bán ở giá cao nhất trong nửa đầu. Lợi nhuận lớn hơn sẽ là kết quả của bài toán.

**Exercise 2.**

Bài toán trên được gọi là bài toán “Unbreakable Laptops”. Bài toán yêu cầu tìm tầng thấp nhất mà chúng ta có thể thả laptop mà không làm vỡ nó. Để giải quyết bài toán này, chúng ta có thể sử dụng phương pháp chia để trị.

Cụ thể, chúng ta sẽ bắt đầu bằng cách thả laptop từ tầng đầu tiên. Nếu laptop không bị vỡ, chúng ta sẽ tiếp tục thả laptop từ các tầng cao hơn cho đến khi laptop bị vỡ. Khi laptop bị vỡ, chúng ta sẽ biết rằng tầng trước đó là tầng cao nhất mà laptop không bị vỡ.

Để giảm số lần thử nghiệm cần thiết, ta sẽ bắt đầu bằng cách thả laptop từ giữa của tầng đầu tiên và tầng cuối cùng. Nếu laptop không bị vỡ, chúng ta sẽ tiếp tục thả laptop từ giữa của các tầng cao hơn cho đến khi laptop bị vỡ. Khi laptop bị vỡ, chúng ta sẽ biết rằng tầng trước đó là tầng cao nhất mà laptop không bị vỡ.

**Phần 3: Bài tập lập trình**

Đánh giá thuật toán qua file exel các bài 1, 2, 4

Demo chương trình trong foder Coding/Hw4/

**Phần 4: Đặt bài toán, thiết kế, phân tích và triển khai thuật toán**

**Ex1. Bài toán tìm cặp điểm gần nhất (Closest pair of point)**

*Phân tích bài toán*

Bài toán tìm cặp điểm gần nhất (closest pair of points) là bài toán tìm hai điểm trong một tập hợp điểm đã cho sao cho khoảng cách giữa chúng là nhỏ nhất.

*Xây dựng thuật toán*

Thuật toán chia để trị (divide and conquer) được sử dụng để giải quyết bài toán tìm cặp điểm gần nhất. Ý tưởng chính của thuật toán là chia tập hợp điểm thành hai phần bằng cách xác định một đường kẻ thẳng (thường là đường dọc) để phân chia các điểm thành hai nhóm. Sau đó, ta sẽ tiếp tục chia các phần con này đến khi không thể chia được nữa, khi đó ta sẽ tìm cặp điểm gần nhất trong từng phần con. Sau đó, ta sẽ so sánh khoảng cách giữa các cặp điểm gần nhất của các phần con để tìm cặp điểm gần nhất của toàn bộ tập hợp điểm.

Các bước chi tiết của thuật toán chia để trị để tìm cặp điểm gần nhất như sau:

1. Sắp xếp các điểm theo trục x.
2. Chia tập hợp điểm thành hai phần bằng đường thẳng trung tâm trên trục x. Nếu số điểm trong tập hợp ban đầu là lẻ thì ta chọn một điểm trung tâm để chia tập hợp thành hai phần bằng nhau.
3. Đệ quy tìm cặp điểm gần nhất trong từng phần con.
4. Tính khoảng cách giữa các cặp điểm gần nhất của từng phần con. Lưu ý rằng ta chỉ cần kiểm tra các cặp điểm có khoảng cách nhỏ hơn khoảng cách giữa hai phần con.
5. So sánh khoảng cách giữa các cặp điểm gần nhất của từng phần con để tìm cặp điểm gần nhất của toàn bộ tập hợp điểm.

*Phân tích thuật toán*

Thời gian chạy của thuật toán chia để trị để tìm cặp điểm gần nhất là O(n log n), với n là số điểm trong tập hợp điểm

*Chương trình mình họa bằng java:*

Foder Coding/hw4/ex4/ClosestPair.java

**Ex2. Phương pháp khử Gauss: giải hệ PT;**

*Phân tích bài toán*

Bài toán đưa ra là giải hệ phương trình tuyến tính bằng phương pháp khử Gauss. Hệ phương trình tuyến tính được biểu diễn dưới dạng ma trận Ax = b, trong đó A là ma trận hệ số, x là vector nghiệm, b là vector hệ số tự do. Phương pháp khử Gauss là một phương pháp giải hệ phương trình tuyến tính bằng cách biến đổi ma trận A thành ma trận tam giác trên bằng cách sử dụng các phép biến đổi dòng. Phương pháp khử Gauss sử dụng kỹ thuật chia để trị để giải quyết bài toán. Kỹ thuật này chia bài toán lớn thành các bài toán nhỏ hơn, giải quyết các bài toán nhỏ đó, và kết hợp các kết quả để tìm ra kết quả cho bài toán ban đầu.

*Xây dựng thuật toán*

1. Biến đổi ma trận A và vector b thành ma trận tam giác trên bằng cách sử dụng các phép biến đổi dòng
2. Giải nghiệm vector x

*Phân tích thuật toán*

Trong bước biến đổi ma trận, thuật toán sử dụng ba vòng lặp lồng nhau để thực hiện các phép biến đổi dòng trên ma trận A và vector b. Độ phức tạp của bước này là O(n^3). Trong bước giải nghiệm vector x, thuật toán sử dụng một vòng lặp để tính toán các giá trị của vector x. Độ phức tạp của bước này là O(n^2). Vì vậy, độ phức tạp của thuật toán khử Gauss là O(n^3), với n là kích thước của ma trận A.

*Chương trình mình họa bằng java:*

Coding/hw4/ex4/Main.java