Cao Thị Thanh Phương – 47.01.104.165

**Bài thực hành 1: Cài đặt và tìm hiểu luật Horner cho đa thức**

Mô tả bài toán:

- Luật Horner được phát biểu như sau:

Với đa thức P(x) =

Ta có thể tính giá trị của đa thức P(x) tại một điểm x bất kỳ theo công thức:

P(x) =

Có thể hiểu đơn giản công thức trên là sử dụng phép tính nhân và cộng lặp lại theo từng hệ số của đa thức.

- Đầu vào:

+ Bậc n của đa thức.

+ Các hệ số a0, a1, ..., an của đa thức.

+ Giá trị x mà ta muốn tính giá trị của đa thức tại đó.

- Đầu ra:

+ Giá trị của đa thức tại x đã cho.

- Cách xử lí:

Để tính giá trị của đa thức theo luật Horner, ta cần thực hiện các bước sau:

+ Khởi tạo biến kq bằng giá trị của hệ số đầu tiên của đa thức, tức là kq = hs[0]

+ Sử dụng vòng lặp for để duyệt qua các hệ số còn lại của đa thức, bắt đầu từ hệ số thứ hai (hs[1]) đến hệ số cuối cùng (hs[n-1]).

+ Trong mỗi lần lặp, cập nhật giá trị kq theo công thức Horner: kq = kq \* x + hs[i].

+ Sau khi vòng lặp kết thúc, kq sẽ chứa giá trị của đa thức tại điểm x đã cho. Hàm trả về giá trị kq là kết quả của phép tính.

Khi tính giá trị của một đa thức tại một điểm x cụ thể bằng cách gọi hàm horner(hs, n, x), trong đó hs là mảng chứa các hệ số của đa thức, n là số lượng hệ số, và x là điểm cần tính giá trị của đa thức

**Bài thực hành 4: Cài đặt quicksort**

Mô tả bài toán:

Thuật toán Quick Sort sẽ tiến hành chia nhỏ mảng thành hai phần. Thông qua phương pháp so sánh từng phần tử với một phần tử chốt, ta sẽ thu được một mảng gồm những phần tử nhỏ hơn hoặc bằng phần tử chốt và một mảng gồm những phần tử lớn hơn phần tử chốt. Hoạt động phân chia này diễn ra liên tục và chỉ dừng lại khi độ dài của mỗi phần tử con bằng 1.

- Đầu vào: Số nguyên n là số lượng phần tử trong mảng và mảng a chứa các số thực cần sắp xếp

- Đầu ra: Mảng a được sắp xếp theo thứ tự tăng dần

- Cách xử lí:

+ Chọn một phần tử x trong mảng a tại vị trí (L + R) / 2 làm phần tử chốt.

+ Khởi tạo hai biến i = L và j = R để chỉ mục bắt đầu và kết thúc của phân đoạn cần xử lí.

+ Thực hiện vòng lặp do-while:

Tìm vị trí đầu tiên i trong phân đoạn từ L đến R sao cho a[i] không nhỏ hơn phần tử chốt x.

Tìm vị trí cuối cùng j trong phân đoạn từ L đến R sao cho a[j] không lớn hơn phần tử chốt x.

Nếu i không vượt qua j, hoán đổi giá trị a[i] và a[j], tăng i và giảm j để tiếp tục tìm kiếm.

+ Lặp lại quá trình trên cho đến khi i vượt qua j.

+Nếu L < j, gọi đệ quy Phanchia(a, L, j) để phân chia và sắp xếp phần đoạn từ L đến j.

+Nếu i < R, gọi đệ quy Phanchia(a, i, R) để phân chia và sắp xếp phần đoạn từ i đến R.

+ Hàm QuickSort: Gọi hàm Phanchia(a, 0, n - 1) để sắp xếp phần đoạn từ 0 đến n-1 của mảng a bằng thuật toán QuickSort.

**Bài thực hành 6: Cài đặt bài toán Tháp Hà Nội**

Mô tả bài toán:

Bài toán tháp Hà Nội là một trò chơi phổ biến. Nó gồm một bộ các đĩa kích thước khác nhau, có lỗ ở giữa, nằm xuyên trên ba cái cột. Bài toán đố bắt đầu bằng cách sắp xếp các đĩa theo trật tự kích thước vào một cột, sao cho đĩa nhỏ nhất nằm ở trên cùng, tức là tạo thành một hình nón.

Ý tưởng:

Nếu đã biết cách chuyển N - 1 đĩa từ cột A sang cột B, ta chỉ cần chuyển đĩa thứ N (đĩa cuối cùng) từ cột A sang cột C, rồi chuyển N - 1 đĩa từ cột B sang cột C.

Giải thuật không còn đệ quy khi chỉ có 1 đĩa, vì ta chuyển trực tiếp từ cột A sang cột C luôn mà không cần thông qua cột B

- Đầu vào: Số lượng đĩa cần di chuyển n

- Đầu ra: Hiện thị các bước di chuyển để chuyển n đĩa từ cột này sang cột kia

- Cách xử lí:

+ Nếu n bằng 1 (đĩa cuối cùng), in ra bước di chuyển đơn giản từ “cot1” sang “cot2”.

+ Ngược lại, thực hiện các bước sau:

Gọi đệ quy TowerOfHanoi(n - 1, cot1, cot3, cot2), di chuyển n-1 đĩa từ “cot1” sang “cot3”, sử dụng “cot2” như cột trung gian.

In ra bước di chuyển đặc biệt để chuyển đĩa n từ “cot1” sang “cot2”.

Gọi đệ quy TowerOfHanoi(n - 1, cot3, cot2, cot1), di chuyển n-1 đĩa từ “cot3” sang“cot2”, sử dụng “cot1” như cột trung gian.

+ Gọi hàm TowerOfHanoi(n, 'A', 'C', 'B') để giải bài toán Tháp Hà Nội với n đĩa, trong đó 'A', 'C', 'B' là tên của 3 cột.