

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT 2

LAB 7 – MỘT SỐ PHƯƠNG PHÁP THIẾT KẾ THUẬT TOÁN (4 TIẾT)

I. Mục tiêu

Sau khi thực hành, sinh viên cần:

- Nắm vững một số phương pháp thiết kế thuật toán: Chia để trị, quay lui, tham lam, quy hoạch động
- Minh họa được bài toán liên quan liên quan đến các phương pháp thiết kế..
- Vận dụng kiến thức đã học để giải một số bài toán thực tế.

II. Yêu cầu

- Sinh viên phải hoàn thành tối thiểu **2 bài** (bài 1 và 1 bài tùy chọn thuộc mục IV). Mỗi bài tạo một project, xóa các thư mục debug của project này. Sau đó chép cả 2 project vào thư mục: Lab7_CTK39_HoTen_MSSV_Nhom#. Nén thư mục, đặt tên tập tin nén theo dạng sau: Lab7_CTK39_HoTen_MSSV_Nhom#.rar.
Ví dụ: Lab7_CTK39_NguyenVanA_141111_Nhom4.rar.
- Sinh viên sẽ nộp bài Lab qua mạng tại phòng lab theo hướng dẫn của giáo viên.

III. Ôn tập lý thuyết

<p>1. Phương pháp chia để trị :</p> <p>Mô hình :</p> <p>Nếu gọi $D\&C(\mathcal{R})$ - Với \mathcal{R} là miền dữ liệu - là hàm thể hiện cách giải bài toán theo phương pháp chia để trị thì ta có thể viết :</p> <pre>void D&C(\mathcal{R}) { If (\mathcal{R} đủ nhỏ) giải bài toán; Else { Chia \mathcal{R} thành $\mathcal{R}_1, \dots, \mathcal{R}_m$; for ($i = 1; i \leq m; i++$) D&C($\mathcal{R}_i$); Tổng hợp kết quả; } }</pre>	<p>2. Phương pháp quay lui</p> <p>Mô hình :</p> <p>Với n là số bước cần phải thực hiện, k là số khả năng mà x_i có thể chọn lựa, $Try(i)$ là bước thử thứ i để xác định x_i</p> <p>$Try(i) \equiv$</p> <pre>for ($j = 1 \rightarrow k$) If (x_i chấp nhận được khả năng j) { Xác định x_i theo khả năng j; Ghi nhận trạng thái mới; if ($i < n$) Try($i+1$); else Ghi nhận nghiệm; Trả lại trạng thái cũ cho bài toán; }</pre>
<p>3. Phương pháp tham lam:</p> <p>Mô hình :</p> <p>Input $A[1..n]$ Output S //lời giải;</p>	<p>4. Phương pháp quy hoạch động:</p> <p>- Phương pháp quy hoạch động dựa vào một nguyên lý, gọi là nguyên lý tối ưu (The principle of optimality) của Bellman : “ Nếu lời giải của bài toán là tối ưu thì lời giải của các bài toán con cũng tối ưu”.</p>

<pre> greedy (A,n) ≡ S = ∅; while (A ≠ ∅) { x= Chọn(A); A = A-{x} if(S ∪ {x} chấp nhận được) S = S ∪ {x}; } return S; </pre>	<p>- Trong thuật toán quy hoạch động thường dùng các thao tác :</p> <ul style="list-style-type: none"> + Xây dựng một hàm quy hoạch động (hoặc phương trình quy hoạch động). + Lập bảng lưu lại các giá trị của hàm. + Truy xuất lời giải tối ưu của bài toán từ bảng lưu.
---	---

IV. Bài tập thực hành

Bài 1: (Chia để trị)

Bài toán Chia thưởng

Bài 2: (Quay lui)

Ngựa đi tuần

Bài 3 : (Phương pháp tham lam)

Bài toán tô màu

Bài 4: (Phương pháp quy hoạch động)

Thuật toán Floy xác định đường đi ngắn nhất giữa các cặp đỉnh

V. Bài tập

Bài 1. Mạng truyền thông

Bài 2. Ông Ngâu, Bà Ngâu.