

BÀI THỰC HÀNH 4

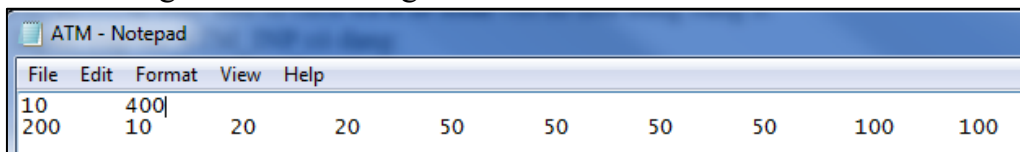
Nội dung:

- Kỹ thuật quay lui
- Kỹ thuật nhánh cận

Bài 4.1 Dùng kỹ thuật quay lui để thực hiện in ra màn hình tất cả các chuỗi nhị phân có độ dài n , với n nhập vào từ bàn phím ($0 < n \leq 10$)

Bài 4.2 Bài toán rút tiền ATM. Một máy ATM hiện có n ($n \leq 20$) tờ tiền với mệnh giá t_1, t_2, \dots, t_n . Hãy đưa ra các cách trả tiền với số tiền đúng bằng S (dùng kỹ thuật quay lui)

- Đầu vào file văn bản (*ATM.inp*) chứa 2 dòng
 - o Dòng đầu chứa giá trị n, s ($n < 30$)
 - o Dòng sau chứa mệnh giá trị của các tờ tiền t_1, t_2, \dots, t_n



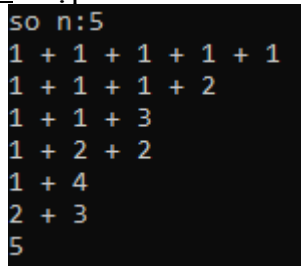
File	Edit	Format	View	Help
10	400			
200	10	20	20	50 50 50 50 100 100

- Đầu ra: file văn bản (*ATM_out.out*) nếu có thể trả đúng s thì đưa ra cách trả, không có cách trả thì ghi -1.

Bài 4.3 Phân tích số

- Hãy tìm tất cả các cách phân tích số n ($0 < n < 20$) thành tổng của các số nguyên dương, các cách phân tích là hoán vị của nhau chỉ tính là 1 cách.
Kết quả phân tích ghi ra file văn bản (*Lab4_3.out*), mỗi cách viết trên 1 dòng; nếu không có cách phân tích thì ghi -1.

Ví dụ: nhập vào $n = 5$



```
so n:5
1 + 1 + 1 + 1 + 1
1 + 1 + 1 + 2
1 + 1 + 3
1 + 2 + 2
1 + 4
2 + 3
5
```

Ý tưởng: Ta sẽ lưu nghiệm trong mảng x , ngoài ra có một mảng s để lưu tổng các phần tử đến bước đang xét. Mảng s xây dựng như sau: s_i sẽ là tổng các phần tử trong mảng x từ x_1 đến x_i : $s_i := x_1 + x_2 + \dots + x_i$.

Khi liệt kê các dãy x có tổng các phần tử đúng bằng n , để tránh sự trùng lặp ta đưa thêm ràng buộc $x_{i-1} \leq x_i$. Khi $s_i = n$ tức là $(x_i = n - s_{i-1})$ thì in kết quả.

Vậy thủ tục Try(i) thử các giá trị cho x_i có thể mô tả như sau: (để tổng quát cho $i = 1$, ta đặt $x_0 = 1$ và $s_0 = 0$).

- Xét các giá trị của x_i từ x_{i-1} đến $(n - s_{i-1})$, cập nhật $s_i := s_{i-1} + x_i$ và gọi đệ quy tìm tiếp.

Bài 4.4* Dùng kỹ thuật quay lui để thực hiện các yêu cầu sau:

a) Liệt kê các tổ hợp chập k của n phần tử

Ví dụ: k = 2, n = 3

(1,2) (1,3) (2,3)

b) Liệt kê các chỉnh hợp không lặp chập k của n

Ví dụ: k = 2, n = 3

(1,2) (1,3) (2,1) (2,3) (3,1) (3,2)

Bài 4.5* Bài toán người du lịch (Nhánh cận)

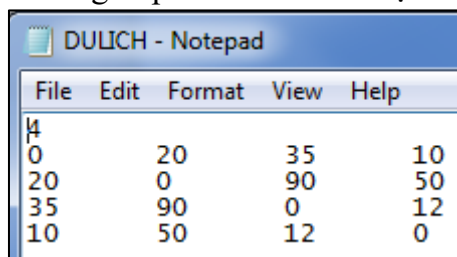
Cho n thành phố đánh số từ 1 đến n và các tuyến đường giao thông hai chiều giữa chúng. Mạng giao thông này được cho bởi mảng hai chiều C, với $C[i][j]=C[j][i]$ là chi phí đi trên đoạn đường nối thành phố i và thành phố j.

Một người du lịch xuất phát từ thành phố 1, muốn đi thăm tất cả các thành phố còn lại mỗi thành phố đúng một lần và cuối cùng về thành phố xuất phát. Hãy chỉ ra hành trình với **chi phí ít nhất**.

- Đầu vào: file văn bản (*Dulich.inp*)

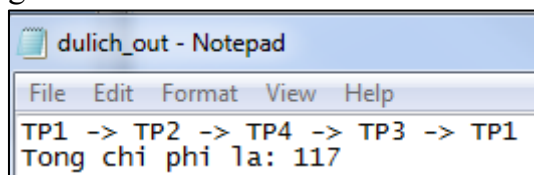
o Dòng đầu chứa số thành phố n ($1 < n < 30$)

o N dòng tiếp theo chứa ma trận chi phí $C[i][j]$



4			
0	20	35	10
20	0	90	50
35	90	0	12
10	50	12	0

- Đầu ra: file văn bản (*Dulich.out*) lưu đường đi với chi phí thấp nhất, không có đường đi thì ghi -1.



TP1 -> TP2 -> TP4 -> TP3 -> TP1
Tổng chi phí là: 117

Gợi ý:

- Hành trình cần tìm có dạng $(x_1 = 1, x_2, \dots, x_n, x_{n+1} = 1)$
- Duyệt quay lui: x_i tiếp theo có thể chọn 1 trong các thành phố chưa đi qua mà từ x_{i-1} có đường đi trực tiếp tới ($2 \leq i \leq n$).
- Nhánh cận: Khởi tạo hành trình tốt nhất Best có chi phí = số có giá trị lớn, Với mỗi bước thử chọn x_i xem chi phí đường đi cho tới lúc đó có nhỏ hơn chi phí của cấu hình Best không? nếu không nhỏ hơn thì thử giá trị khác ngay. Khi thử được một giá trị x_n ta kiểm tra xem x_n có đường đi trực tiếp về 1 không? Nếu có đánh giá chi phí đi từ thành phố 1 đến thành phố x_n cộng với chi phí từ x_n đi trực tiếp về 1, nếu nhỏ hơn chi phí của đường đi Best thì cập nhật lại Best bằng cách đi mới.

----- hết -----