Quay Lui

Nhánh-Cận

1. Quay lui(Back Tracking)

-1 số lưu ý :Tránh SD nhiều vòng for.

1. Tư tưởng:

-Bài toán yêu cầu nghiệm được biểu diễn dưới dạn vector x[1]…x[k] .

-Trong đó x[i] là 1 phần tử được lấy ra từ tập S{}.

-Xét lần lượt các khả năng của x[i] lấy từ tập S.

-Các bước trong việc liệt kê cấu hình dạng X[1...n]:

+Xét tất cả các giá trị X[1] có thể nhận, thử X[1] nhận các giá trị đó. Với mỗi giá trị của X[1] ta sẽ:

+Xét tất cả giá trị X[2] có thể nhận, lại thử X[2] cho các giá trị đó. Với mỗi giá trị X[2] lại xét khả năng giá trị của X[3]...tiếp tục như vậy cho tới bước:

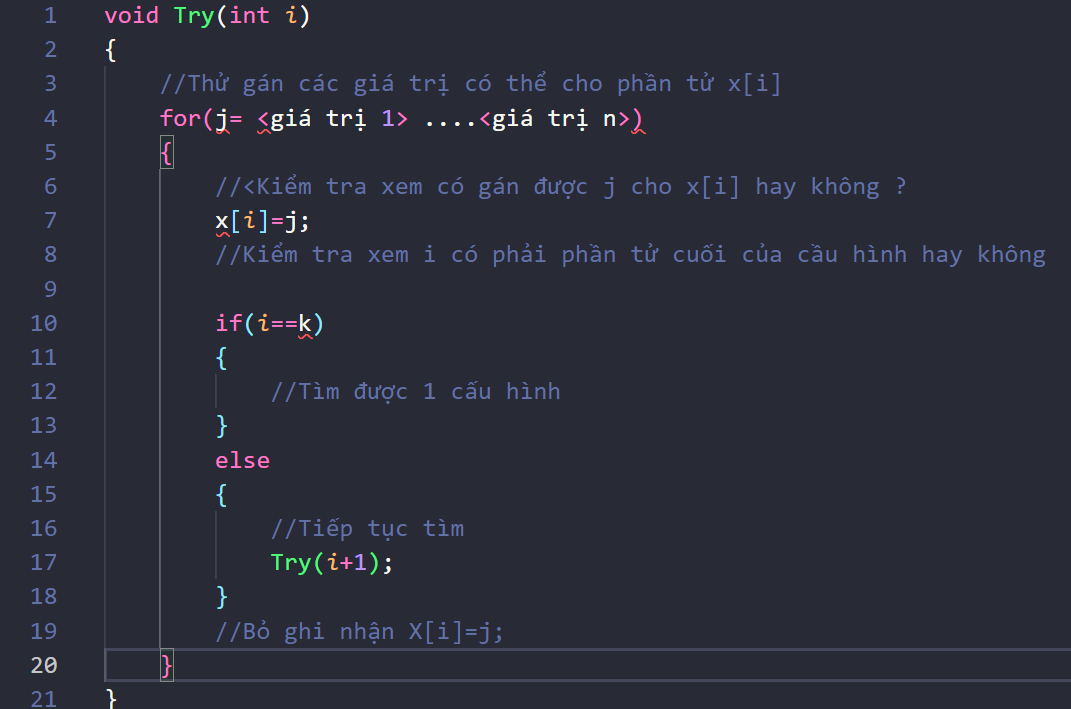
...

...

+Xét tất cả giá trị X[n] có thể nhận, thử cho X[n] nhận lần lượt giá trị đó.

+Thông báo cấu hình tìm được.

1. Mô hình thuật toán quay lui.



1. Ưu,nhược điểm:

-Ưu điểm: Chạy qua hết tất cả các cấu hình hoàn chỉnh,tránh được các cấu hình chưa hoàn chỉnh

-Nhược điểm:Trường hợp xấu nhất là cấp số mũ.Vì nó mắc phải nhược điểm sau:

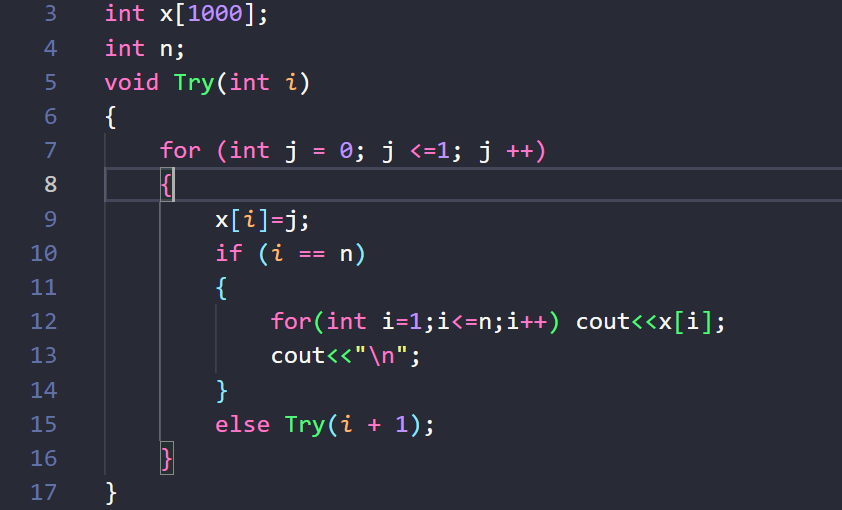
+Quá trình tìm kiếm cứ gặp phải bế tắc với cùng một nguyên nhân.

+Thực hiện các công việc dư thừa: Mỗi lần chúng ta quay lui, chúng ta cần phải đánh giá lại lời giải trong khi đôi lúc điều đó không cần thiết.

+Không sớm phát hiện được các khả năng bị bế tắc trong tương lai. Quay lui chuẩn, không có cơ chế nhìn về tương lai để nhận biết đc nhánh tìm kiếm sẽ đi vào bế tắc.

VD : Sinh Nhị Phân: X[I] sẽ có 2 khả năng :=0 hoặc =1.

=> Vậy ta chỉ cần cho x[I] chọn 0 hoặc chọn 1 và đi tiếp tới x[I+1]

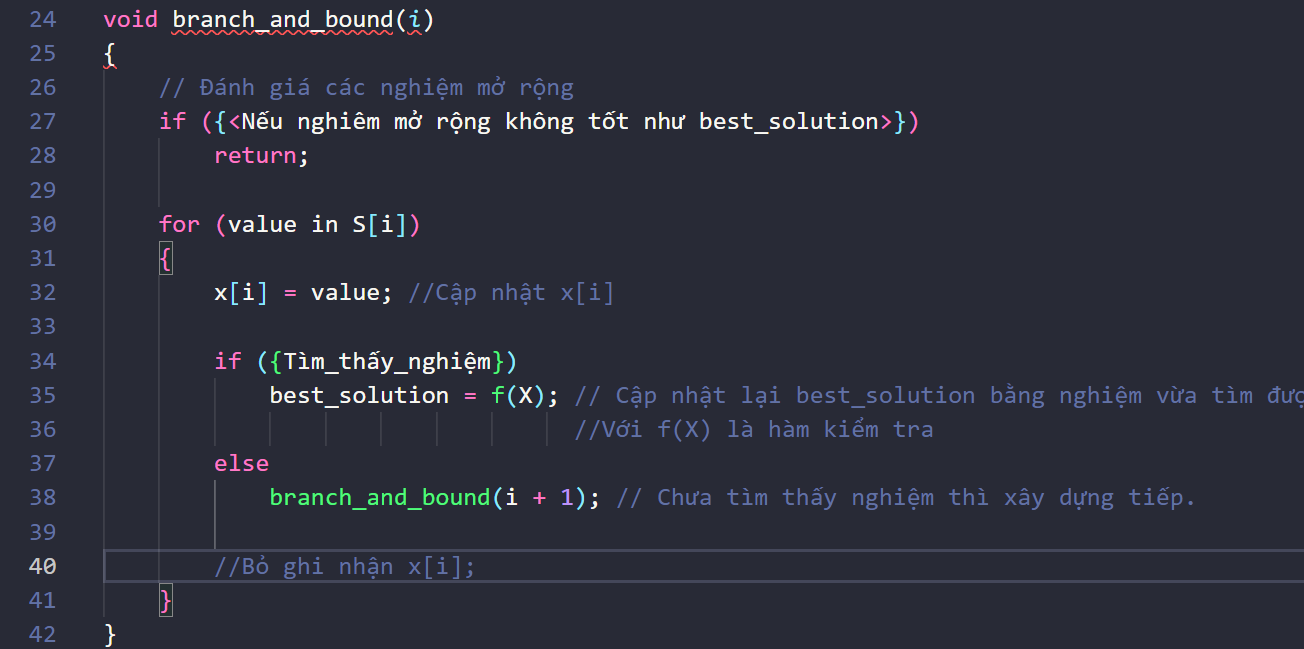


Code: <https://ideone.com/LJJJJv>

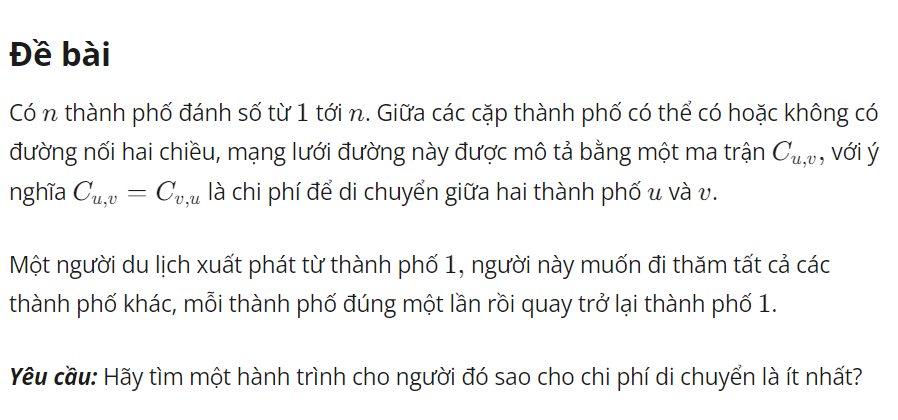
1. Nhánh và Cận (Branch and Bound)
2. Giới thiệu: Là 1 phương pháp cải tiến từ phương pháp quay lui.Chỉ khác là với mỗi cấu hình đang xét,ta sẽ có thêm 1 bước “Đánh giá độ tối ưu” bằng 1 hàm f(X)

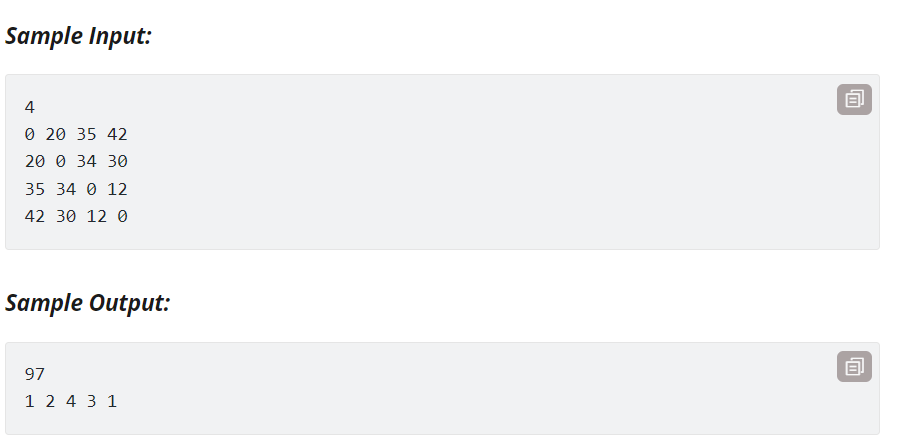
=> Vậy ta chỉ cần tìm được hàm f(X) tối ưu là xong.

1. Sơ đồ chung:

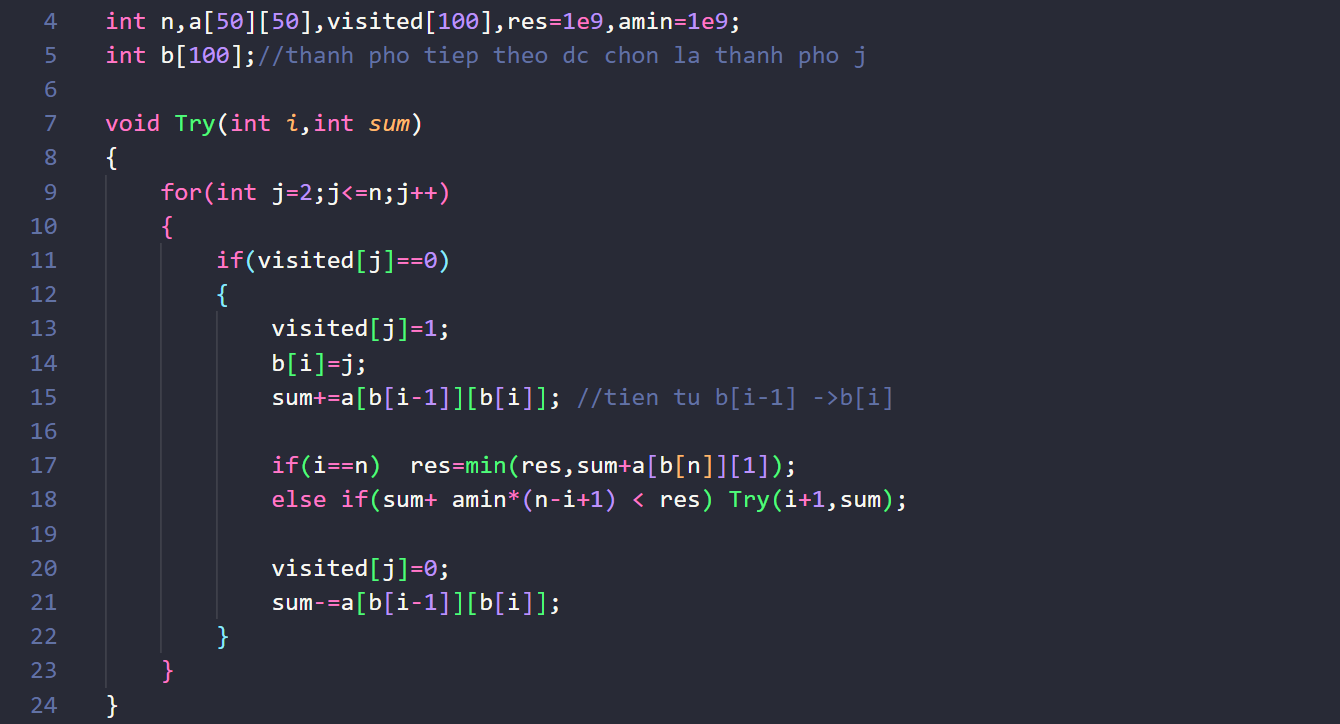


1. Bài toán tham khảo:





-Quay lui :



Code: <https://ideone.com/dFZbT7>

-Trong phần code trên amin là số tiền nhỏ nhất khi đi từ thành phố I-> thành phố J bất kì.

-Khi ta đi đến thành phố thứ i,tổng tiền đã dùng = Sum

-Ta còn n-i+1 thành phố nữa cần đến (Tính cả thành phố 1) tương ứng với n-I+1 quãng cần đi

-Vậy tổng tiền cần >= Sum + amin\*(n-I+1).

Mà nếu Sum + amin\*(n-I+1) > Res (Phương án tối ưu nhất đang xét) => Không cần xét cấu hình này nữa