



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

THUẬT TOÁN ỨNG DỤNG

Tìm kiếm quay lui

Sơ đồ chung tìm kiếm quay lui

- Áp dụng để giải các bài toán liệt kê, bài toán tối ưu tổ hợp
- $A = \{(x_1, x_2, \dots, x_n) \mid x_i \in A_i, \forall i = 1, \dots, n\}$
- Liệt kê tất cả các bộ $x \in A$ thoả mãn một thuộc tính P nào đó
- Thủ tục TRY(k):
 - Thử các giá trị v có thể gán cho x_k mà không vi phạm thuộc tính P
 - Với mỗi giá trị hợp lệ v :
 - Gán v cho x_k
 - Nếu $k < n$: gọi đệ quy TRY($k+1$) để thử tiếp giá trị cho x_{k+1}
 - Nếu $k = n$: ghi nhận cấu hình

Sơ đồ chung tìm kiếm quay lui

```
TRY( $k$ )
  Begin
    Foreach  $v$  thuộc  $A_k$ 
      if check( $v, k$ ) /* kiểm tra xem  $v$  có hợp lệ không */
        Begin
           $x_k = v$ ;
          [Cập nhật các cấu trúc dữ liệu D đi kèm]
          if( $k = n$ ) ghi_nhan_cau_hinh;
          else TRY( $k+1$ );
          [Khôi phục D]
        End
      End
    End
  End

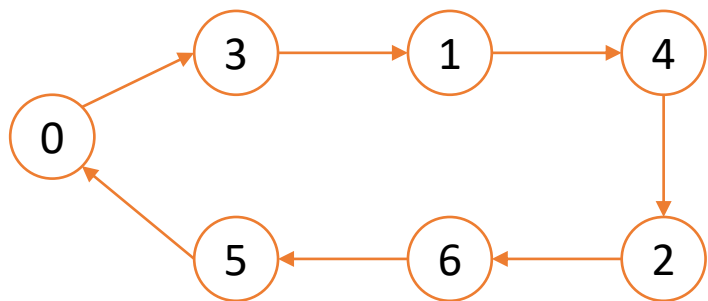
Main()
  Begin
    TRY(1);
  End
```

Bài toán lập lộ trình xe buýt

- Một xe buýt xuất phát từ điểm 0, cần phục vụ yêu cầu đón trả của n khách $1, 2, \dots, n$, trong đó khách i có điểm đón là i và điểm trả là $i + n$. Xe buýt có Q chỗ ngồi cho khách. Khoảng cách từ điểm i đến điểm j là $d(i, j)$, với mọi $i, j = 0, 1, \dots, 2n$. Hãy lập lộ trình cho xe buýt xuất phát từ điểm 0, thực hiện đón trả n khách và quay về điểm 0 sao cho tổng quãng đường di chuyển là nhỏ nhất.

Bài toán lập lộ trình xe buýt

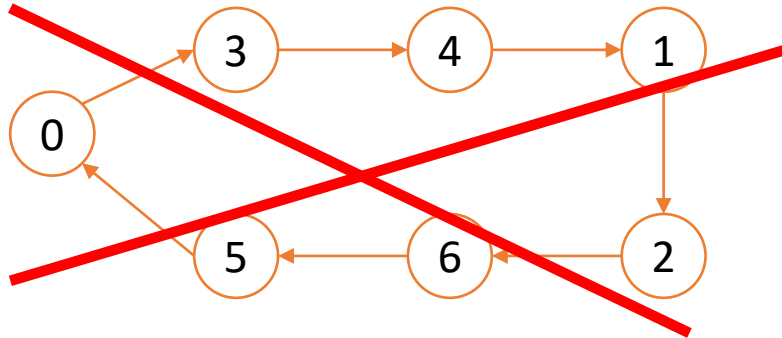
- Một xe buýt xuất phát từ điểm 0, cần phục vụ yêu cầu đón trả của n khách 1, 2, ..., n , trong đó khách i có điểm đón là i và điểm trả là $i + n$. Xe buýt có Q chỗ ngồi cho khách. Khoảng cách từ điểm i đến điểm j là $d(i, j)$, với mọi $i, j = 0, 1, \dots, 2n$. Hãy lập lộ trình cho xe buýt xuất phát từ điểm 0, thực hiện đón trả n khách và quay về điểm 0 sao cho tổng quãng đường di chuyển là nhỏ nhất.



3	2					
0	8	5	1	10	5	9
9	0	5	6	6	2	8
2	2	0	3	8	7	2
5	3	4	0	3	2	7
9	6	8	7	0	9	10
3	8	10	6	5	0	2
3	4	4	5	2	2	0

Bài toán lập lộ trình xe buýt

- Một xe buýt xuất phát từ điểm 0, cần phục vụ yêu cầu đón trả của n khách 1, 2, ..., n , trong đó khách i có điểm đón là i và điểm trả là $i + n$. Xe buýt có Q chỗ ngồi cho khách. Khoảng cách từ điểm i đến điểm j là $d(i, j)$, với mọi $i, j = 0, 1, \dots, 2n$. Hãy lập lộ trình cho xe buýt xuất phát từ điểm 0, thực hiện đón trả n khách và quay về điểm 0 sao cho tổng quãng đường di chuyển là nhỏ nhất.

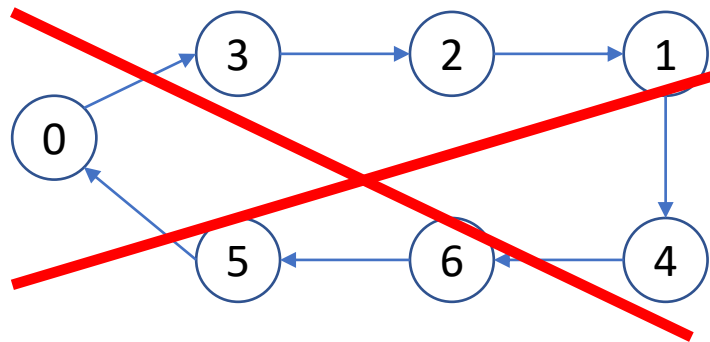


Vi phạm ràng buộc
thứ tự đón-trả

3	2					
0	8	5	1	10	5	9
9	0	5	6	6	2	8
2	2	0	3	8	7	2
5	3	4	0	3	2	7
9	6	8	7	0	9	10
3	8	10	6	5	0	2
3	4	4	5	2	2	0

Bài toán lập lộ trình xe buýt

- Một xe buýt xuất phát từ điểm 0, cần phục vụ yêu cầu đón trả của n khách 1, 2, ..., n , trong đó khách i có điểm đón là i và điểm trả là $i + n$. Xe buýt có Q chỗ ngồi cho khách. Khoảng cách từ điểm i đến điểm j là $d(i, j)$, với mọi $i, j = 0, 1, \dots, 2n$. Hãy lập lộ trình cho xe buýt xuất phát từ điểm 0, thực hiện đón trả n khách và quay về điểm 0 sao cho tổng quãng đường di chuyển là nhỏ nhất.



Vi phạm ràng buộc
sức chứa

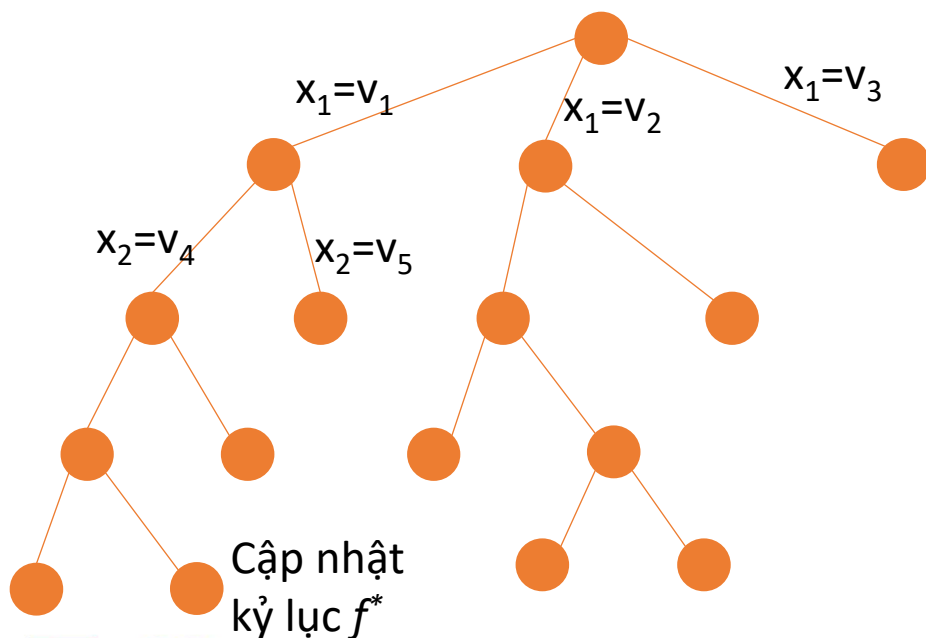
3	2					
0	8	5	1	10	5	9
9	0	5	6	6	2	8
2	2	0	3	8	7	2
5	3	4	0	3	2	7
9	6	8	7	0	9	10
3	8	10	6	5	0	2
3	4	4	5	2	2	0

Bài toán lập lộ trình xe buýt: tổ chức dữ liệu

- $x[0, \dots, 2n]$: mô hình hóa lộ trình, trong đó
 - $x[0] = 0$ là điểm xuất phát,
 - Lộ trình là $x[0] \rightarrow x[1] \rightarrow x[2] \rightarrow \dots \rightarrow x[2n] \rightarrow x[0]$
- f : độ dài của lộ trình bộ phận (được cập nhật tăng dần trong quá trình duyệt)
- f_best : ghi nhận độ dài lộ trình ngắn nhất
- $load$: ghi nhận số khách đang có mặt trên xe trong lộ trình bộ phận
- $visited[1, \dots, 2n]$, trong đó $visited[v] = true$ có nghĩa điểm v đã được đi qua và $visited[v] = false$, nếu ngược lại

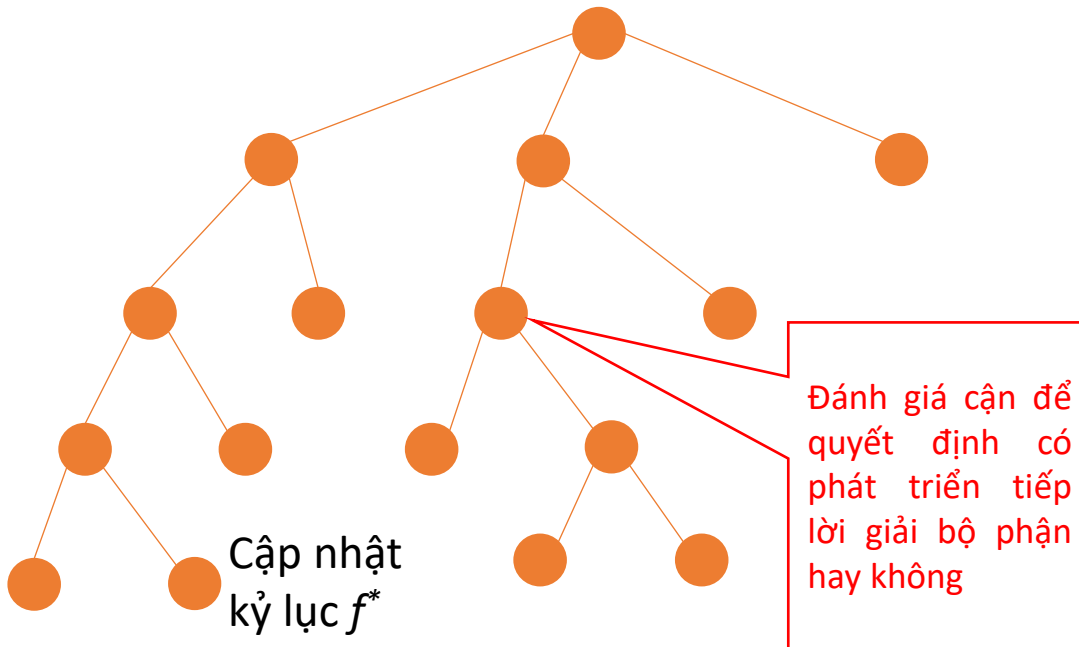
Thuật toán nhánh và cận

- Duyệt toàn bộ:
 - Liệt kê tất cả các phương án bằng phương pháp đệ quy quay lui
 - Với mỗi phương án, tính toán hàm mục tiêu
 - Giữ lại phương án có hàm mục tiêu nhỏ nhất



```
TRY( $k$ )
Begin
  Foreach  $v$  thuộc  $A_k$ 
    if check( $v, k$ )
      Begin
         $x_k = v$ ;
        [cập nhật  $f$ , load, visited]
        if( $k = 2n$ )
          ghi_nhan_cau_hinh;
          cập nhật kỷ lục  $f_{best}$ ;
        else TRY( $k+1$ );
        [khôi phục  $f$ , load, visited]
      End
    End
  End
Main()
Begin
  TRY(1);
End
```

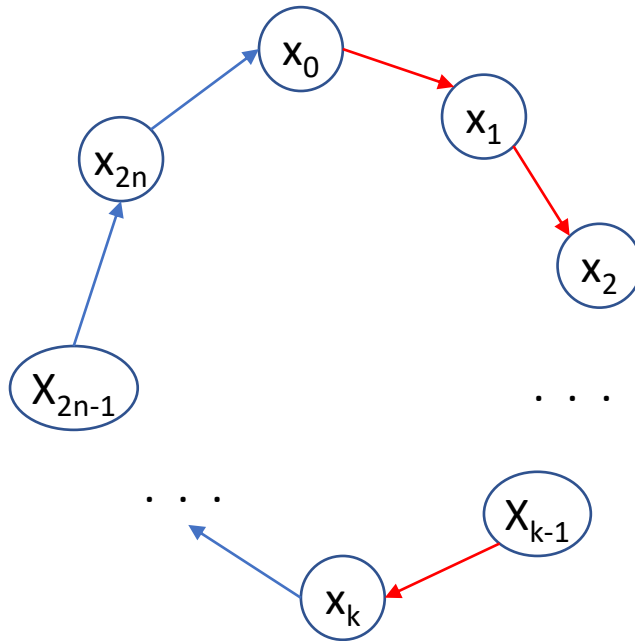
Thuật toán nhánh và cận



```
TRY( $k$ ) {  
    Foreach  $v$  thuộc  $A_k$   
        if check( $v, k$ ) {  
             $x_k = v$ ;  
            [cập nhật  $f$ , load, visited]  
            if( $k = 2n$ ) {  
                ghi_nhan_cau_hinh;  
                cập nhật kỷ lục  $f\_best$ ;  
            }else{  
                if  $g(x_1, \dots, x_k) < f\_best$   
                    TRY( $k+1$ );  
            }  
            [khôi phục  $f$ , load, visited]  
        }  
    }  
Main(){  
     $f\_best = \infty$ ;  
    TRY(1);  
}
```

Thuật toán nhánh và cận

- d_m là khoảng cách nhỏ nhất trong số các khoảng cách đi giữa 2 thành phố khác nhau
- Phương án bộ phận (x_0, \dots, x_k)
 - Chi phí bộ phận $f = d(x_0, x_1) + d(x_1, x_2) + \dots + d(x_{k-1}, x_k)$
 - Hàm cận dưới : $g(x_0, \dots, x_k) = f + d_m \times (2n - k + 1)$





25 YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**



soict.hust.edu.vn/



fb.com/groups/soict

