BRANCH AND B UND



- ☐ Các khái niệm
- ☐ Các thuật toán phổ biến
- ☐ Các bài toán
- ☐ Bài tập về nhà

Nhóm 07:

- Mai Đại Lâm
- Nguyễn Phú Quang

Thế nào là **branch and bound**?

- Một phường pháp thiết kế thuật toán.
- Thường được sử dụng để giải bài toán tối ưu tổ hợp.
- Duyệt các nhánh trên cây nhưng dựa vào một cận (giới hạn được ta đưa vào để giảm thiểu số lần duyệt).
- Nếu nhánh tiếp theo là trường hợp tệ hơn của cận đang có ta loại nó khỏi bài toán đang xét.

Đặc điểm của một bài toán tối ưu tổ hợp?

- Bao gồm một tổ hợp các vấn đề nhỏ
- Tìm lời giải tối ưu trong các lời giải khả thi
- Kết quả cuối cùng là tổ hợp các lời giải con tối ưu

Các cách tiếp cận Branch and bound

- FIFO Branch and bound
- LIFO Branch and bound
- LC Branch and bound

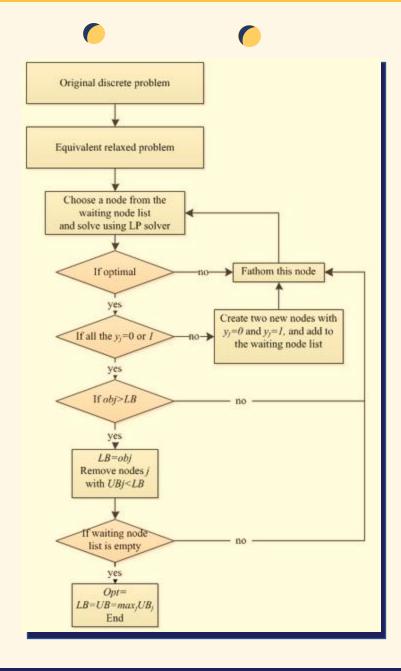
1.Các khái niệm			

Branch and bound vs. Backtracking

Bản thân Branch and bound chính là Backtracking **nhưng** ta xác định một cận **(chi phí ước lượng)** để loại bỏ các nhánh không chứa lời giải.

Khái niệm	Backtracking	Branch and Bound
Cách tiếp cận	Tìm mọi trường khả thi bằng duyệt cây. Lùi lại 1 bước nếu lựa chọn sai.	Dùng giải quyết bài toán tối ưu hóa. Nếu đã có hướng giải quyết tốt hơn thì không tiếp tục hướng hiện tại
Bài toán giải quyết	Decisions Problem	Optimisation Problem

1.Các khái niệm



2. Dạng thuật toán tổng quát			

First in first out - Branch and bound

- Khởi tạo một giá trị Upperbound(B) cho bài toán. Nếu không có cơ sở đặt $B=\infty$. Giá trị này biểu diễn kết quả tốt nhất hiện tại
- Khởi tạo hàng đợi rỗng
- Coi cây có gốc là E và sinh ra các node con cho E.
- Đưa vào hàng đợi các node con E.

Ở mỗi vòng lặp:

- Loại node đầu tiên khỏi hàng đợi sinh các node con cho node đó r đưa nó vào hàng đợi
- ❖ Kết quả **N** cho đến node hiện tại
 - **N** < **B** thì ta đặt **B** = **N**.
 - N > B thì ta không xét tiếp node con của node hiện tại.

2. Dạng thuật toán tổng quát				

Last in first out - Branch and bound

- Khởi tạo một giá trị Upperbound(B) cho bài toán. Nếu không có cơ sở đặt $B=\infty$. Giá trị này biểu diễn kết quả tốt nhất hiện tại
- Khởi tạo stack rỗng
- Coi cây có gốc là E và sinh ra các node con cho E.
- Đưa vào hàng đợi các node con E.

Ở mỗi vòng lặp:

- Loại node cuối cùng khỏi stack sinh các node con cho node đó r đưa nó vào hàng đợi
- ❖ Kết quả **N** cho đến node hiện tại
 - **N** < **B** thì ta đặt **B** = **N**.
 - N > B thì ta không xét tiếp node con của node hiện tại.

2. Dạng thuật toán tổng quát				

Lowest cost-Branch and bound

- Khởi tạo một giá trị Upperbound(B) cho bài toán. Nếu không có cơ sở đặt $B=\infty$. Giá trị này biểu diễn kết quả tốt nhất hiện tại
- Khởi tạo hàng đợi rỗng
- Coi cây có gốc là E và sinh ra các node con cho E.
- Đưa vào hàng đợi các node con E.
- Ở mỗi vòng lặp:
 - Chỉ đi sinh ra các node con cho node có chi phí thấp nhất.

2. Dạng thuật toán tổng quát			

Uu/Nhược điểm

Ưu điểm: Nếu sử dụng cách duyệt dựa trên node tối ưu nhất thì độ phức tạp về thời gian sẽ nhỏ hơn các thuật toán khác

Đây là một kĩ thuật hoạt động tốt cho dữ liệu kích thước nhỏ. Nếu kích thước dữ liệu quá lớn có thể dẫn tới việc nhánh được chọn duyệt không tối ưu. Từ đó dẫn tới:

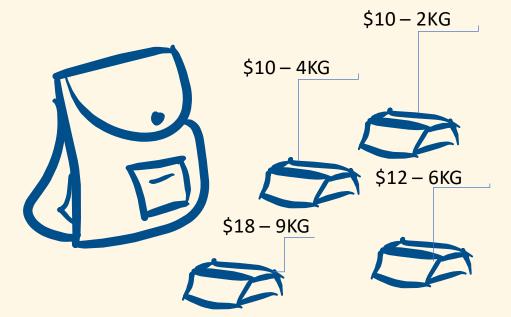
Nhược điểm: Tốn thời gian cho bộ dữ liệu có kích thước quá lớn.

2. Dạng thuật toán tông quát				

2. Dạng thuật toán tổng quát Một số bài toán có thể giải sử dụng BnB □ 0/1 Knapsack ☐ Job assignment problem. ☐ Traveling Salesman problem. ☐ 8 Puzzle problem. ☐ N Queen problem.

0/1 Knapsack

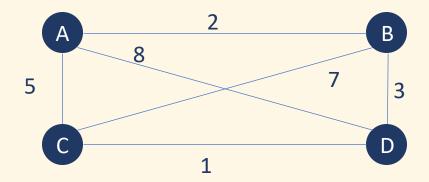
- Cho N món đồ với cân nặng và giá trị riêng
- Tìm tập hợp các món hàng để bỏ vào balo sao cho tổng giá trị là lớn nhất và tổng cân nặng không vượt quá giá trị W = 15



2. Các bài toán

Travelling Salesman

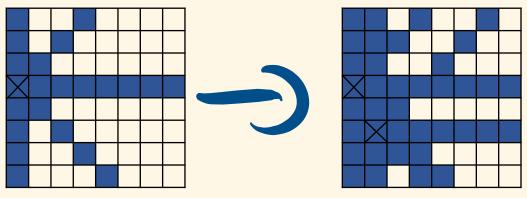
- Một người đi giao hàng tại *N* địa điểm.
- Người đó chỉ được đi qua tất cả các địa điểm 1
 lần và trở về vị trí ban đầu.
- Khoảng cách giữa các địa điểm đã được biết trước.
- Tìm một chu trình sao cho tổng độ dài là nhỏ nhất.

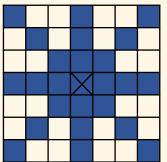


2. Các bài toán

N - Queens

Trong một bàn cờ vua N x N đặt vào các quân hậu vào bàn cờ sao cho không quân hậu nào nằm trên trường đi của bất kì quân hậu nào khác





2. Các bài toán				

participating