**Bài toán cây bao trùm tối thiểu**

1. **Phương pháp nhánh cận**
2. **Giới thiệu**

* Phương pháp nhánh cận là một kỹ thuật trong lĩnh vực tối ưu hóa và giải thuật, được sử dụng để tìm kiếm một giải pháp tối ưu hoặc gần tối ưu cho các bài toán tối ưu hóa.

1. **Đánh giá thuật toán**
2. Ý tưởng

* Nhánh cận (Branch and Bound): Thuật toán tìm lời giải cho các bài toán tối ưu dạng liệt kê cấu hình dựa trên nguyên lí đánh giá nhánh cận.
* Nguyên lí đánh giá nhánh cận: Sử dụng các thông tin đã tìm được trong lời giải của bài toán để loại bỏ sớm phương án không dẫn tới lời giải tối ưu
* Bản chất:

+ Sử dụng phương pháp quay lui nhưng tại mỗi bước đưa thêm thao tác đánh giá giá trị phương án hiện có.

+ Nếu đó là phương án tối ưu hoặc có hy vọng trở thành phương án tối ưu (tức là tốt hơn phương án hiện có) thì cập nhật lại phương án tối ưu hoặc đi tiếp theo hướng đó.

+ Trong trường hợp ngược lại thì bỏ qua hướng đang xét.

1. Mã giả
2. function BranchAndBoundMST(graph):
3. MST = InitializeMST(graph)  // Khởi tạo MST ban đầu chứa một đỉnh bất kỳ
5. queue.push(MST)  // Đưa MST vào hàng đợi
7. **while** queue **is** **not** empty:
8. current = queue.pop()  // Lấy MST từ hàng đợi
10. **if** current **is** a complete MST:
11. **if** current.weight < bestMST.weight:
12. bestMST = current  // Cập nhật MST tối thiểu tốt nhất
13. **else**:
14. **for** each edge **in** graph:
15. **if** edge **is** **not** **in** current:
16. newMST = current.clone() // Sao chép MST hiện tại để tạo MST mới
18. **if** newMST.canAdd(edge):
19. newMST.addEdge(edge)  // Thêm cạnh vào MST mới
21. **if** newMST.lowerBound < bestMST.weight:  // Kiểm tra giới hạn
22. queue.push(newMST)  // Đưa MST mới vào hàng đợi
24. **return** bestMST

* Sử dụng một hàng đợi (queue) để lưu trữ các MST tiềm năng. Quá trình nhánh cận tiếp tục cho đến khi không còn MST nào trong hàng đợi. Nếu một MST được coi là hoàn chỉnh (complete MST), tức là chứa tất cả các đỉnh của đồ thị, chúng ta so sánh trọng số của nó với MST tối thiểu tốt nhất (bestMST) hiện tại và cập nhật nếu cần.
* Nếu MST không hoàn chỉnh, chúng ta thêm các cạnh chưa được thêm vào MST hiện tại để tạo ra các MST mới. Trước khi thêm cạnh mới, chúng ta kiểm tra xem nó có thể được thêm vào MST hay không, và kiểm tra giới hạn (lower bound) của MST mới. Nếu giới hạn đó nhỏ hơn trọng số của MST tối thiểu tốt nhất, chúng ta đưa MST mới vào hàng đợi để tiếp tục quá trình nhánh cận.
* Cuối cùng, trả về MST tối thiểu tìm được (bestMST).

1. Độ phức tạp

* Độ phức tạp thuật toán phụ thuộc vào 2 yếu tố:

+ Số đỉnh của đồ thị: Khi đồ thị có n đỉnh, phương pháp nhánh cận có thể tạo ra nhiều đồ thị con khác nhau trong quá trình nhánh cận. Do đó, số lượng đồ thị con và các phép toán tương ứng sẽ tăng theo cấp số nhân với số đỉnh của đồ thị.

+ Trọng số của các cạnh: Độ phức tạp của phép so sánh trọng số các cạnh trong quá trình nhánh cận cũng ảnh hưởng đến độ phức tạp của thuật toán. Nếu trọng số của các cạnh là nguyên, thì việc so sánh trọng số có thể được thực hiện với độ phức tạp O(1). Tuy nhiên, nếu trọng số của các cạnh là số thực hoặc số phức, thì việc so sánh trọng số có thể có độ phức tạp cao hơn.

1. Chứng minh tính đúng đắn