

# Môn: Nhập môn Trí tuệ Nhân tạo

- Bài tập thực hành tuần: 5
- Sinh viên: Huỳnh Thị Bảo Trân
- MSSV: 19110482

## Câu hỏi:

**4.8** The traveling salesperson problem (TSP) can be solved via the minimum spanning tree (MST) heuristic, which is used to estimate the cost of completing a tour, given that a partial tour has already been constructed. The MST cost of a set of cities is the smallest sum of the link costs of any tree that connects all the cities.

- Show how this heuristic can be derived from a relaxed version of the TSP.
- Show that the MST heuristic dominates straight-line distance.
- Write a problem generator for instances of the TSP where cities are represented by random points in the unit square.
- Find an efficient algorithm in the literature for constructing the MST, and use it with an admissible search algorithm to solve instances of the TSP.

## Bài làm:

(a). TSP có thể được định nghĩa là một khách du lịch phải đến thăm  $x$  thành phố, nơi mỗi thành phố chỉ được đến thăm một lần và họ kết thúc ở thành phố ban đầu của họ, với mục tiêu giảm thiểu thời gian di chuyển. MST là cùng một vấn đề với sự thư giãn mà các thành phố có thể được đến thăm nhiều hơn một lần và chi phí để đi trên một con đường đã đi là 0.

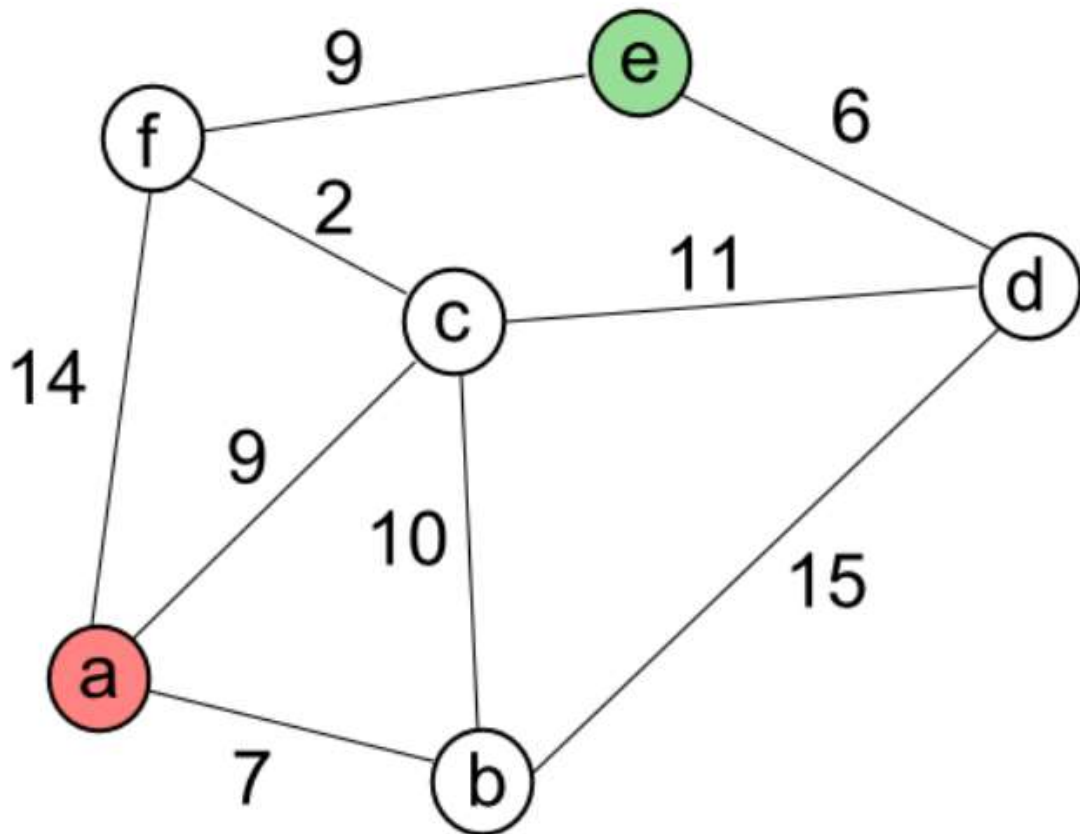
(b) MST chiếm ưu thế về khoảng cách đường thẳng vì khoảng cách ngắn nhất từ hai thành phố  $u, v$  là khoảng cách đường thẳng của chúng. Hơn nữa, cho  $GMST(V, E^-)$  là nghiệm của bài toán MST cho ví dụ  $G$ . Chi phí từ  $u$  đến  $v$  cho MST sẽ bằng khoảng cách đường thẳng chỉ khi  $(u, v) \in E^-$ . Do đó, heuristic MST chi phối khoảng cách đường thẳng. MST là đường thẳng vì nó dựa trên khoảng cách kết nối tối thiểu, làm cho nó trở thành một phương pháp dò tìm có thể chấp nhận được.

(d) Trong lý thuyết của độ phức tạp thuật toán, phiên bản quyết định của bài toán TSP thuộc lớp NP-complete. Vì vậy không có giải thuật hiệu quả nào cho việc giải bài toán TSP. Hay nói cách khác, thời gian xấu nhất cho giải thuật cho bài toán TSP tăng theo hàm mũ với số lượng thành phố, vì vậy trường hợp xấu nhất với vài trăm thành phố thì mất khoảng vài năm để có thể giải bài toán một cách chính xác.

(c) Trình bày giải pháp giải bài toán.

In [ ]:

```
# giả sử em chọn graph để thực hiện như hình
from IPython.display import display, Image
display(Image(filename='graph.png'))
```



In [ ]:

```

from sys import maxsize
from itertools import permutations
V = 4

def TSP(graph, s):

    # Lưu trữ tất cả các đỉnh ngoài đỉnh gốc
    vertex = []
    for i in range(V):
        if i != s:
            vertex.append(i)

    # Lưu trữ trọng lượng tối thiểu Chu kỳ Hamilton
    minpath = maxsize
    nextper = permutations(vertex)
    for i in nextper:

        # Lưu trữ trọng số đường dẫn hiện tại (chi phí)
        current = 0

        # tính toán trọng lượng đường dẫn hiện tại
        k = s
        for j in i:
            current += graph[k][j]
            k = j
        current += graph[k][s]
  
```

```
# cập nhật
minpath = min(minpath, current)

return minpath

graph = [[0, 7, 9, 0, 0, 14],
          [7, 0, 10, 15, 0, 0],
          [9, 10, 0, 11, 0, 2],
          [0, 15, 11, 0, 6, 0],
          [0, 0, 0, 6, 0, 9],
          [14, 0, 2, 0, 9, 0]]
print('cost is', TSP(graph, 0))
```

cost is 28