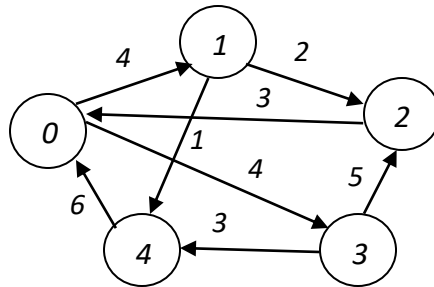


Ôn tập

Cho đồ thị G như sau:



A. Tạo dữ liệu:

1. Tạo file MTK.txt biểu diễn cho đồ thị trên.
2. Tạo file DSK.txt biểu diễn cho đồ thị trên.

B. Chuyển đổi cấu trúc

1. Viết hàm chuyển Ma trận kề sang danh sách kề

`AdjacencyList MTK_to_DSK(AdjacencyMatrix MTKe)`

2. Viết hàm chuyển Ma trận kề sang danh sách cạnh

`ArrayEdge MTK_to_DSC(AdjacencyMatrix MTKe)`

3. Viết hàm chuyển danh sách kề sang ma trận kề

`AdjacencyMatrix DSK_to_MTK(AdjacencyList DSKe)`

4. Viết hàm chuyển danh sách kề sang danh sách cạnh

`ArrayEdge DSK_to_DSC(AdjacencyList DSKe)`

5. Viết hàm chuyển danh sách cạnh và 1 số nguyên là số đỉnh của đồ thị sang ma trận kề

`AdjacencyMatrix DSC_to_MTK(ArrayEdge DSC, int sodinh)`

6. Viết hàm chuyển danh sách cạnh và 1 số nguyên là số đỉnh của đồ thị sang danh sách kề

`AdjacencyList DSC_to_DSK(ArrayEdge DSC, int sodinh)`

C. Tính Bậc

1. Viết hàm tính bậc vào của 1 đỉnh (Danh sách kề).

`int BacVao(AdjacencyList DSKe, VERTEX u)`

2. Viết hàm tính bậc ra của 1 đỉnh u (Danh sách kề).

`int TinhBacRa(AdjacencyList DSKe, VERTEX u);`

3. Viết hàm tính bậc vào của 1 đỉnh (Ma trận kề).

`int BacVao(AdjacencyMatrix x, VERTEX u)`

4. Viết hàm tính bậc ra của 1 đỉnh u (Ma trận kề).

`int TinhBacRa(AdjacencyMatrix x, VERTEX u);`

5. Viết hàm trả về đỉnh có bậc lớn nhất trong đồ thị (Ma trận kề).
`VERTEX DinhBacMax(AdjacencyMatrix x)`
6. Viết hàm trả về đỉnh có bậc nhỏ nhất trong đồ thị (Ma trận kề).
`VERTEX DinhBacMin(AdjacencyMatrix x)`
7. Viết hàm trả về đỉnh có bậc lớn nhất trong đồ thị (Danh sách kề).
`VERTEX DinhBacMax(AdjacencyList x)`
8. Viết hàm trả về đỉnh có bậc nhỏ nhất trong đồ thị (Danh sách kề).
`VERTEX DinhBacMin(AdjacencyList x)`
9. Viết hàm tìm đỉnh v kề đỉnh u có bậc lớn nhất (Ma trận kề). Hàm trả về 1 nếu tìm được và trả về 0 nếu không tìm được.

`int TimDinhKe_BacMax(AdjacencyMatrix MTKe, VERTEX u, VERTEX &v)`

10. Viết hàm tìm đỉnh v kề đỉnh u có bậc nhỏ nhất (Ma trận kề)

`int TimDinhKe_BacMin(AdjacencyMatrix MTKe, VERTEX u, VERTEX &v)`

11. Viết hàm tìm đỉnh v kề đỉnh u có bậc lớn nhất (Danh sách kề). Hàm trả về 1 nếu tìm được và trả về 0 nếu không tìm được.

`int TimDinhKe_BacMax(AdjacencyList DSKe, VERTEX u, VERTEX &v)`

12. Viết hàm tìm đỉnh v kề đỉnh u có bậc nhỏ nhất (Danh sách kề). Hàm trả về 1 nếu tìm được và trả về 0 nếu không tìm được.

`int TimDinhKe_BacMin(AdjacencyList DSKe, VERTEX u, VERTEX &v)`

D. Tìm Cạnh

1. Viết hàm tìm cạnh lớn nhất trong đồ thị (Ma trận kề), hàm trả về 1 nếu có cạnh lớn nhất, trả về 0 nếu không có cạnh lớn nhất

`int TimCanhMax(AjgencyMatrix MTKe, EDGE &CanhMax)`

2. Viết hàm tìm cạnh nhỏ nhất trong đồ thị (Ma trận kề), hàm trả về 1 nếu có cạnh nhỏ nhất, trả về 0 nếu không có cạnh nhỏ nhất

`int TimCanhMin(AjgencyMatrix MTKe, EDGE &CanhMin)`

3. Viết hàm tìm cạnh lớn nhất trong đồ thị (Danh sách kề), hàm trả về 1 nếu có cạnh lớn nhất, trả về 0 nếu không có cạnh lớn nhất

`int TimCanhMax(AdjacencyList DSKe, EDGE &CanhMax)`

4. Viết hàm tìm cạnh nhỏ nhất trong đồ thị (Danh sách kề), hàm trả về 1 nếu có cạnh nhỏ nhất, trả về 0 nếu không có cạnh nhỏ nhất

int TimCanhMin(AdjacencyList DSKe, EDGE &CanhMin)

5. Viết hàm trả về tổng trọng số của các cạnh trong đồ thị (Ma trận kề)

int TongTrongSo(AdjacencyMatrix MTKe)

6. Viết hàm trả về tổng trọng số của các cạnh trong đồ thị (Danh sách kề)

int TongTrongSo(AdjacencyList DSKe)

7. Viết hàm trả về tổng trọng số các cạnh xuất phát từ đỉnh u (ma trận kề)

int TongTrongSoTuU(AdjacencyMatrix MTKe, VERTEX u)

8. Viết hàm trả về tổng trọng số các cạnh đi vào đỉnh u (ma trận kề)

int TongTrongSoVaoU(AdjacencyMatrix MTKe, VERTEX u)

9. Viết hàm trả về tổng trọng số các cạnh xuất phát từ đỉnh u (danh sách kề)

int TongTrongSoTuU(AdjacencyList DSKe, VERTEX u)

10. Viết hàm trả về tổng trọng số các cạnh đi vào đỉnh u (danh sách kề)

int TongTrongSoVaoU(AdjacencyList DSKe, VERTEX u)

11. Viết hàm tìm cạnh có trọng số lớn nhất kề với đỉnh u (ma trận kề). Hàm trả về 1 nếu tìm được, trả về 0 nếu không tìm được.

int CanhCoTrongSoLonNhat(AdjacencyMatrix x, int u, EDGE &CanhKeMax)

12. Viết hàm tìm cạnh có trọng số lớn nhất kề với đỉnh u (danh sách kề). Hàm trả về 1 nếu tìm được, trả về 0 nếu không tìm được.

int CanhCoTrongSoLonNhat(AdjacencyList x, int u, EDGE &CanhKeMax)

13. Viết hàm tìm cạnh có trọng số nhỏ nhất kề với đỉnh u (ma trận kề). Hàm trả về 1 nếu tìm được, trả về 0 nếu không tìm được.

int CanhCoTrongSoNhoNhat(AdjacencyMatrix x, int u, EDGE &CanhKeMin)

14. Viết hàm tìm cạnh có trọng số nhỏ nhất kề với đỉnh u (danh sách kề). Hàm trả về 1 nếu tìm được, trả về 0 nếu không tìm được.

int CanhCoTrongSoNhoNhat(AdjacencyList x, int u, EDGE &CanhKeMin)

15. Viết hàm tìm đỉnh v kề với đỉnh u có trọng số cạnh (u,v) nhỏ nhất (Ma trận kề). Hàm trả về 1 nếu tìm được, trả về 0 nếu không tìm được.

int TimDinhke_MinTrongso(AdjacencyMatrix MTKe, VERTEX u, VERTEX &v)

16. Viết hàm tìm đỉnh v kề với đỉnh u có trọng số cạnh (u,v) lớn nhất (Ma trận kề).
Hàm trả về 1 nếu tìm được, trả về 0 nếu không tìm được.

```
int TimDinhke_MaxTrongso(AdjacencyMatrix MTKe, VERTEX u, VERTEX &v)
```

17. Viết hàm tìm đỉnh v kề với đỉnh u có trọng số cạnh (u,v) nhỏ nhất (Danh sách kề).
Hàm trả về 1 nếu tìm được, trả về 0 nếu không tìm được.

```
int TimDinhke_MinTrongso(AdjacencyList DSKe, VERTEX u, VERTEX &v)
```

18. Viết hàm tìm đỉnh v kề với đỉnh u có trọng số cạnh (u,v) lớn nhất (Danh sách kề).
Hàm trả về 1 nếu tìm được, trả về 0 nếu không tìm được.

```
int TimDinhke_MaxTrongso(AdjacencyList DSKe, VERTEX u, VERTEX &v)
```