* برای استفاده از پکیج نتورک ایکس در وی اس کد باید اول افزونه ی پایتون را در وی اس کد نصب کنیم سپس برنامه اصلی پایتون را در سیستم خود نصب میکنیم حال برای شروع کار از منوی فایل یک فایل با پسوند .py ایجاد میکنیم.
* برای شروع کار باید دو پکیج را به پروژه اضافه کنیم .

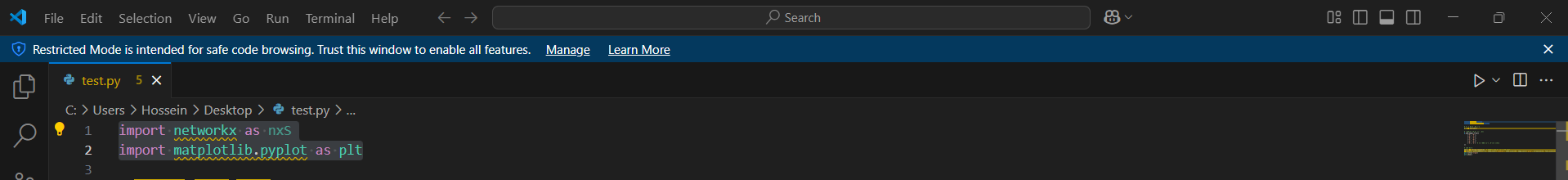
1: network X

2: matplotlib.pyplot

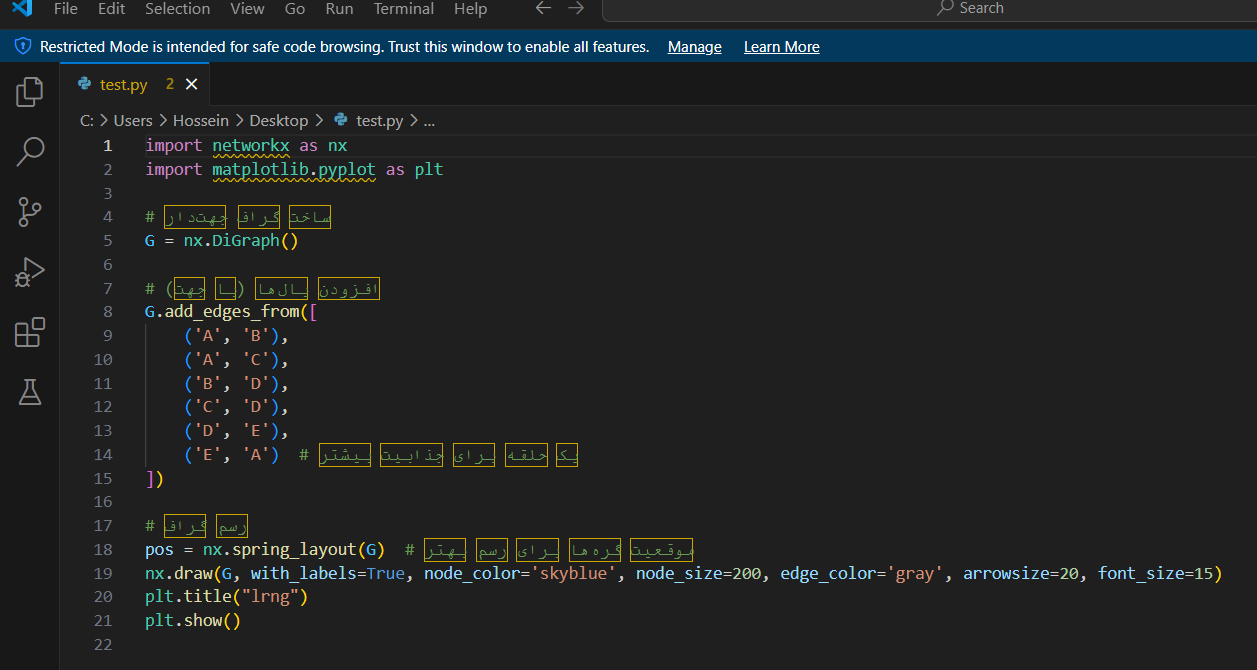
* ترمینال را از منوی ترمینال باز میکنیم و دستور pip install networkx اجرا میکنیم .
* حال فایل امده برای نوشتن دستورات هست ابتدا این دو پکیج را اضافه میکنیم .

import networkx as nxS

import matplotlib.pyplot as plt

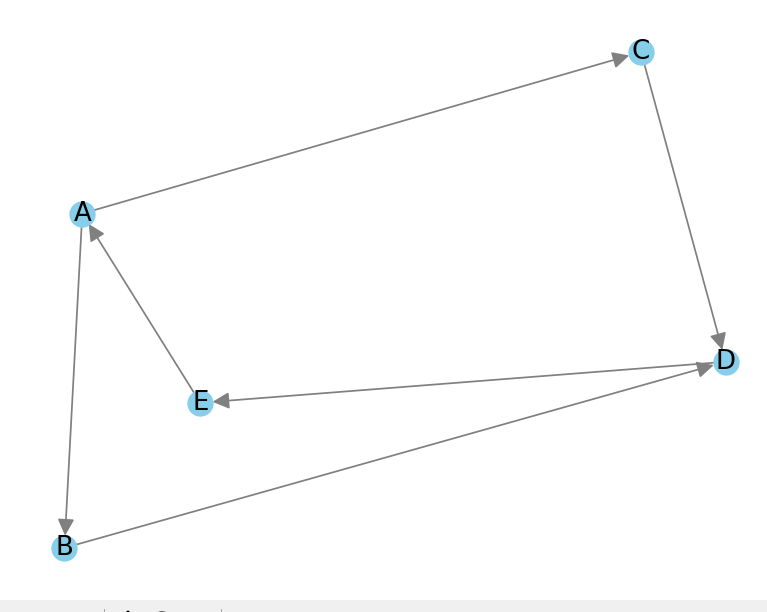


* حال یک گراف جهت دار رسم میکنیم(ساده).

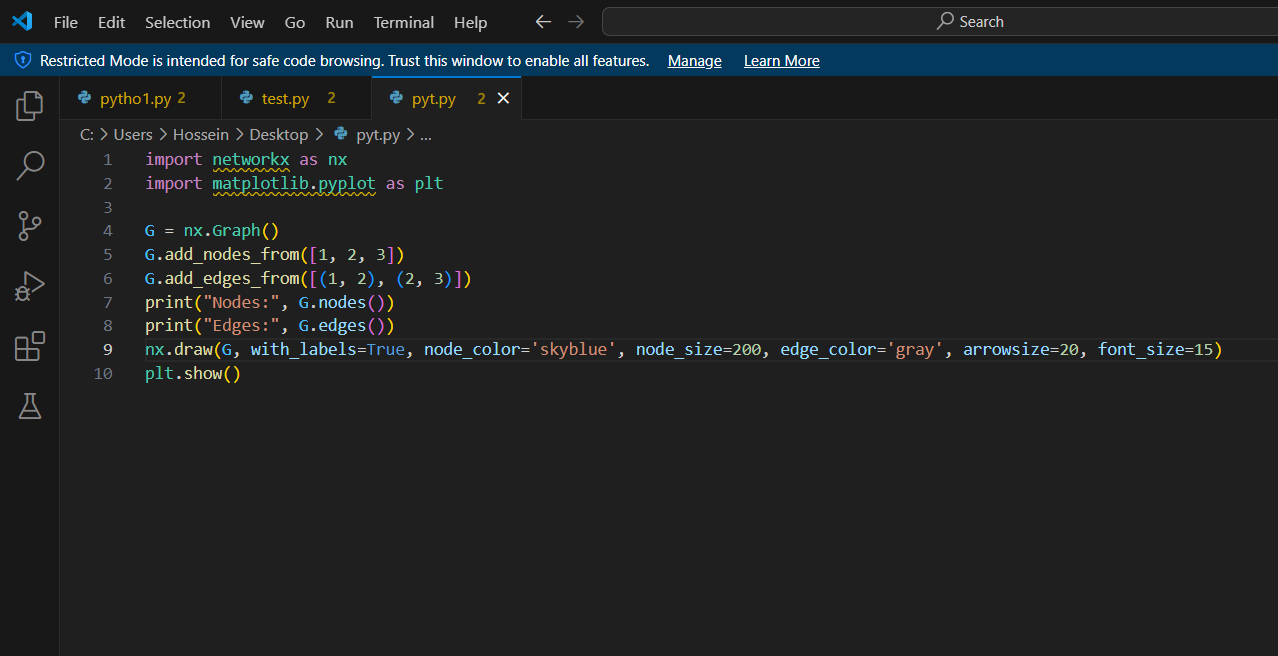


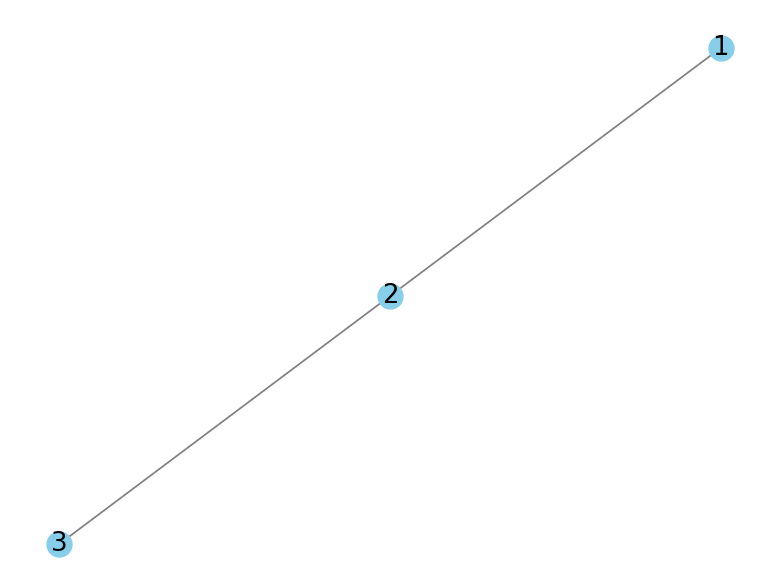
* با ترمینال به محل ذخیره فایل میرویم و با دستور python test.py برنامه را اجرا میکنیم .

.

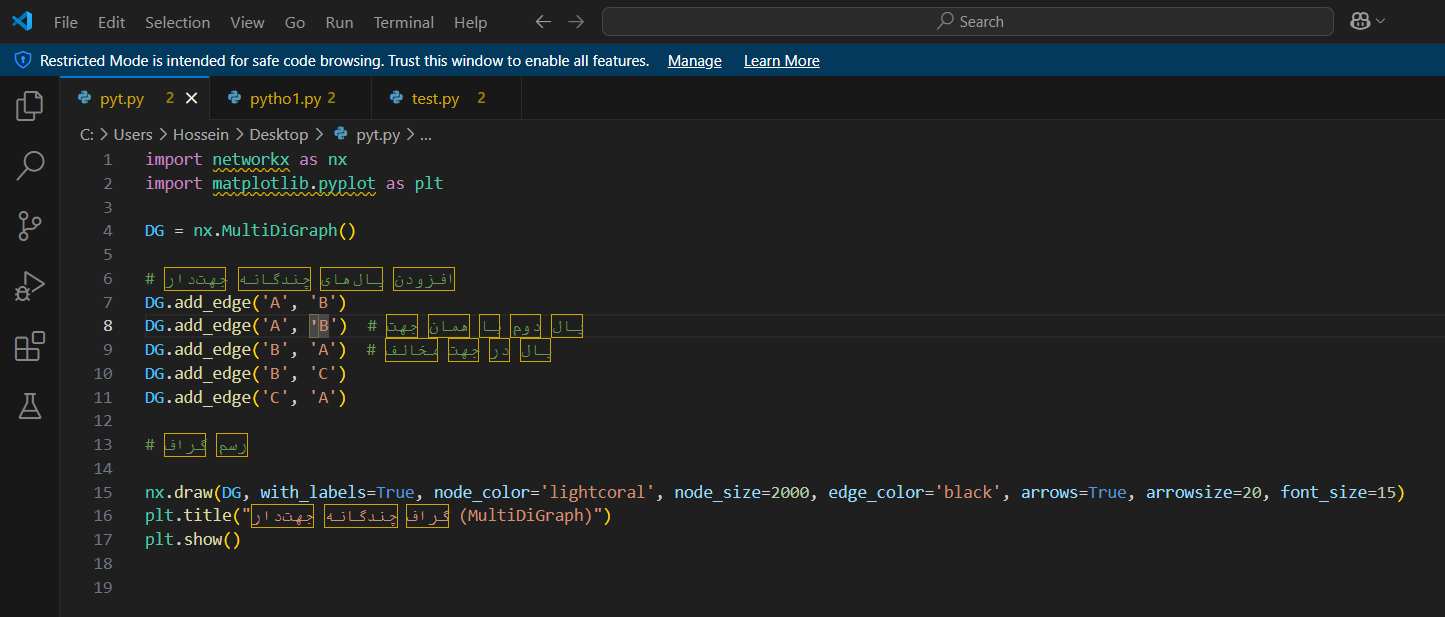


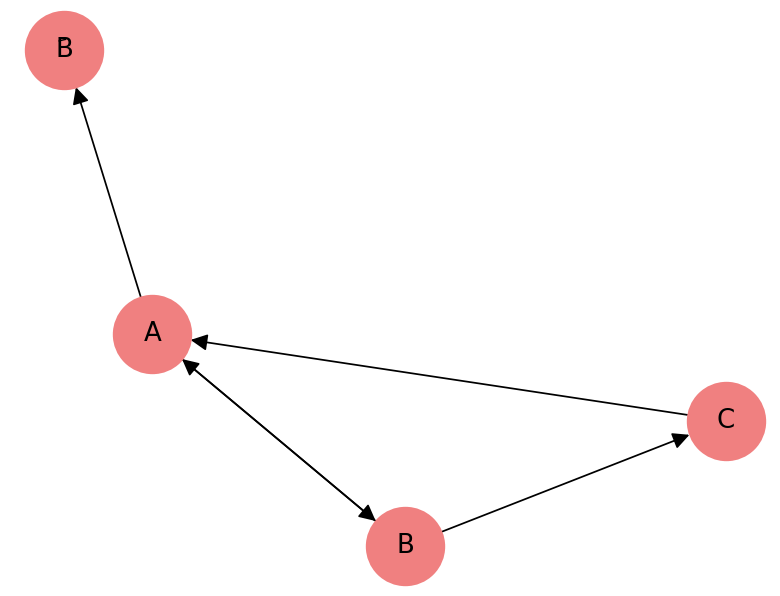
* گراف بدونه جهت (ساده) .>>

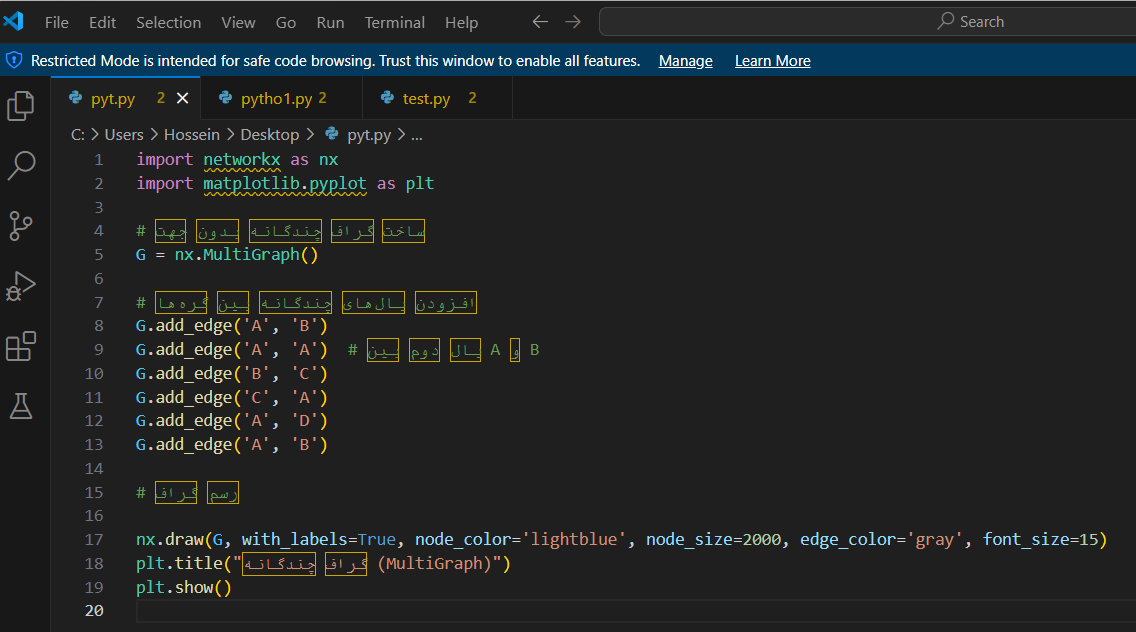


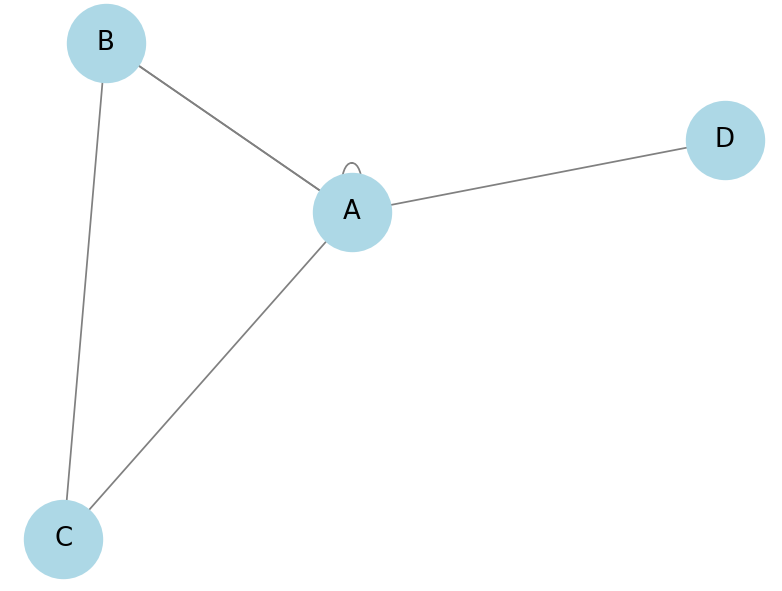


* گراف چند گانه(multigraph): یعنی گرافی که بین دو گره می‌تونه چند یال(با جهت یا بدون جهت) داشته باشه.

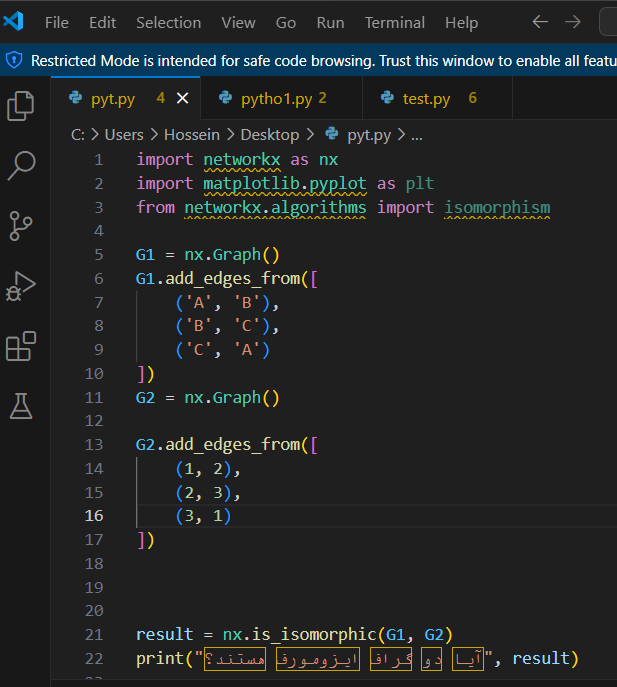






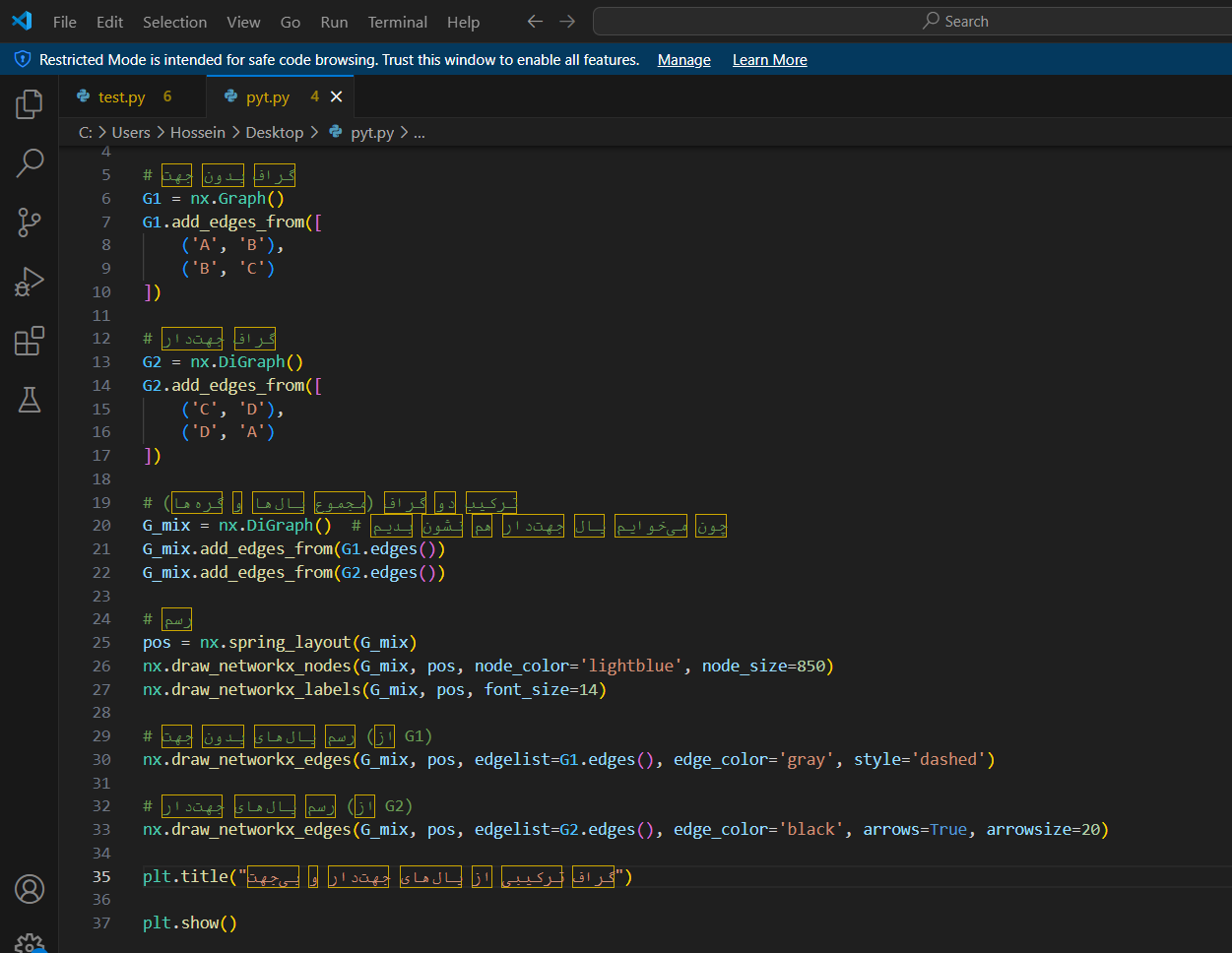


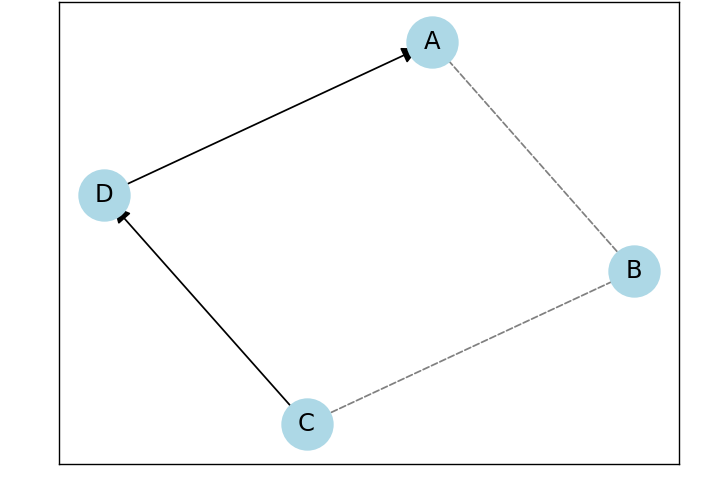
* گراف ایزومورفیک (Isomorphic Graphs) : ساختار آن‌ها یکسان باشد، یعنی فقط نام گره‌ها فرق کند، ولی اتصال بین گره‌ها (یال‌ها) دقیقاً یکسان باشد



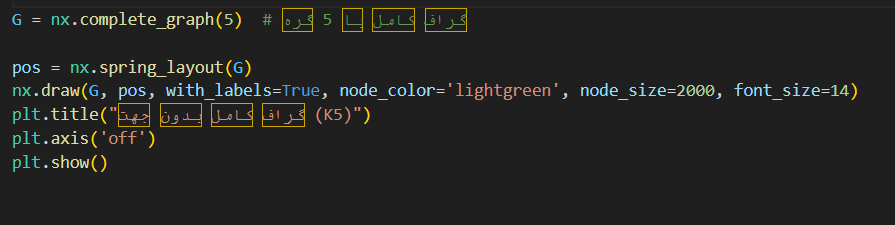
خروجی true و false

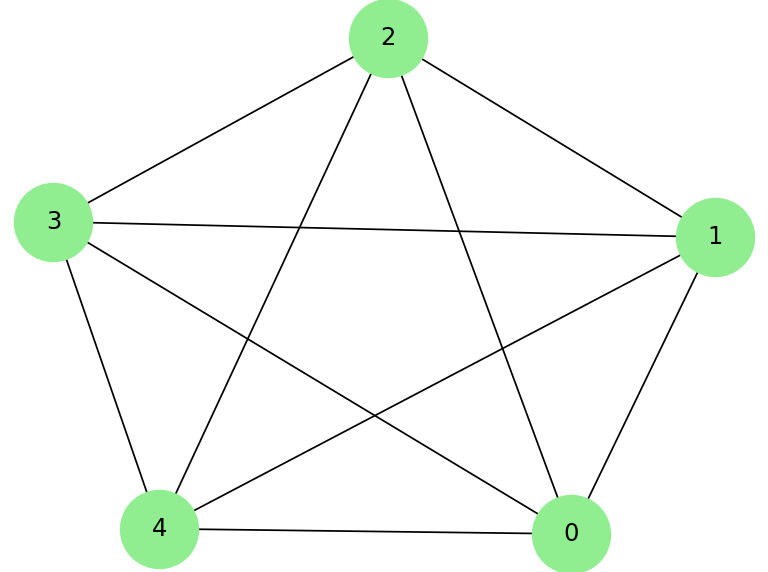
* گراف میکس : گراف ترکیبی از جهت‌دار و بدون جهت



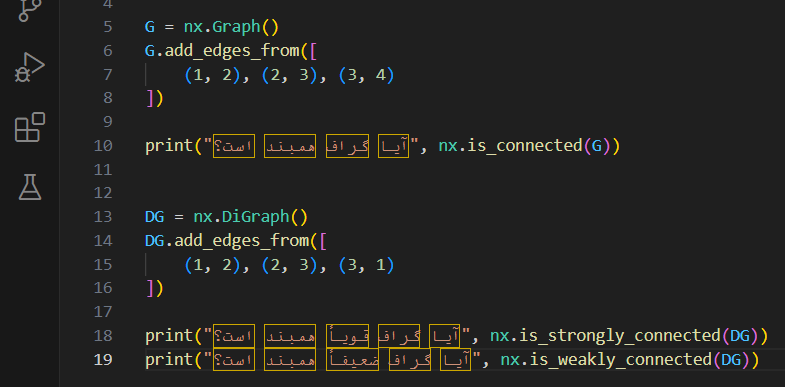


* گراف کامل : بین هر دو گره یک یال وجود دارد.



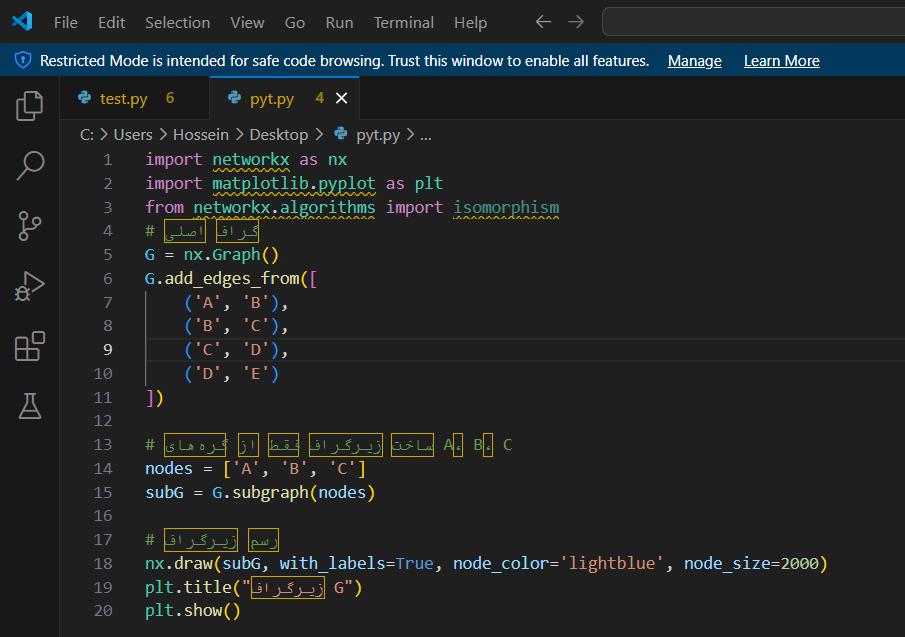


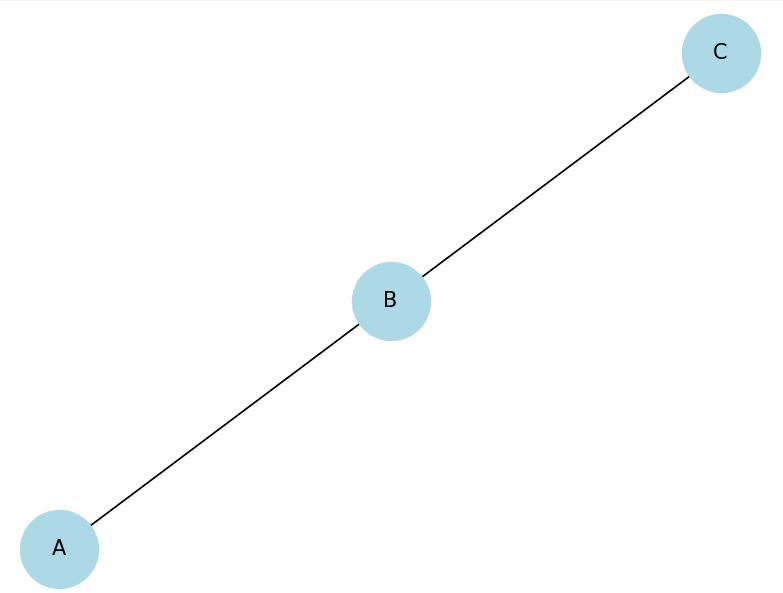
هم بند بودن با نبودن : بین هر دو گره آن، حداقل یک مسیر وجود دارد.



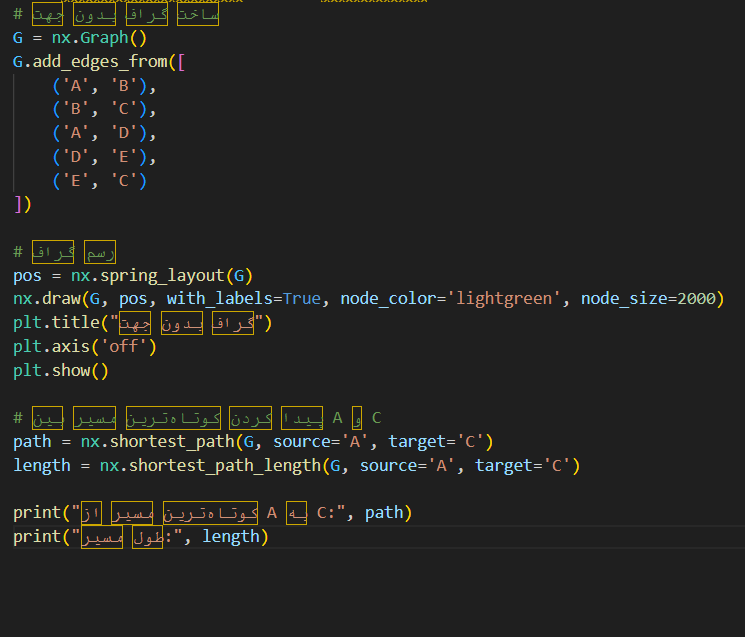
خروجی true و false

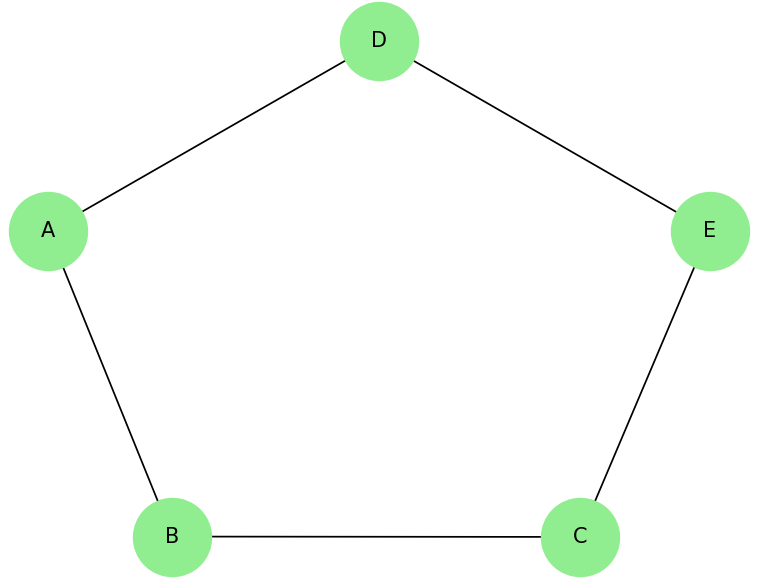
* زیر گراف : یک بخش از گراف اصلی که شامل بعضی از گره‌ها و یال‌های بین آن‌ها است.



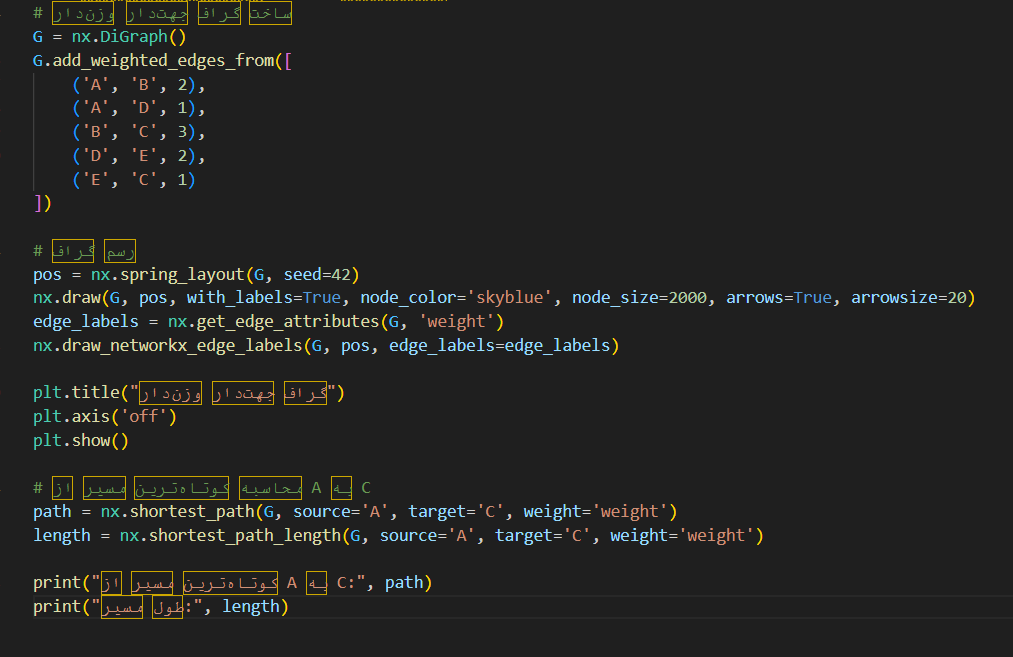


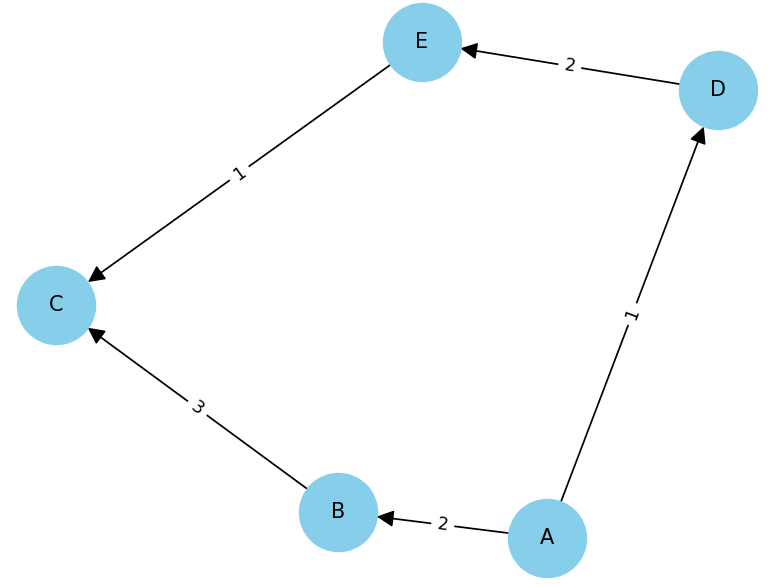
* کوتاه ترین مسیر در گراف بدونه جهت (برابر بودن وزن همه یال ها)



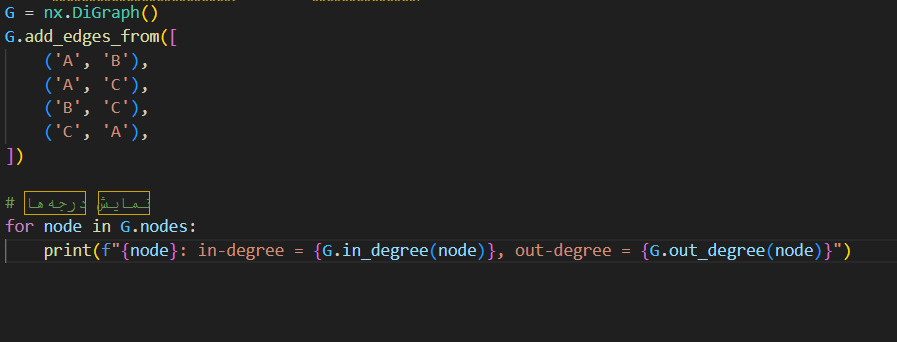


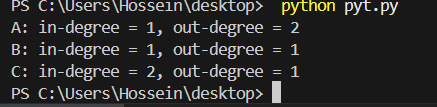
کوتاه‌ترین مسیر در گراف جهت‌دار (Directed Graph): یعنی مسیری با کمترین تعداد یال (در گراف بدون وزن) یا کمترین مجموع وزن (در گراف وزن‌دار)، با رعایت جهت‌ها.



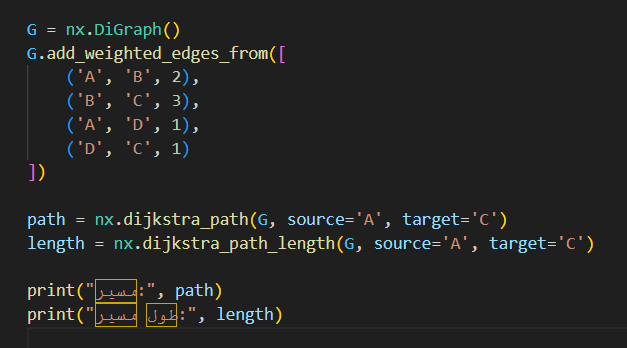


* درجه رأس (Degree of a Node): تعداد یال‌هایی که به آن متصل است.



خروجی : 

* الگوریتم دایکسترا (Dijkstra’s Algorithm) : پیدا کردن کوتاه‌ترین مسیر از یک گره به بقیه گره‌ها در گراف وزن‌دار بدون وزن منفی.



* مسیر اویلری (Eulerian Path): مسیری که هر یال گراف دقیقاً یک‌بار طی بشه (ممکنه گره‌ها تکرار بشن)
* مدار اویلری (Eulerian Circuit) : مسیری که هر یال دقیقاً یک‌بار طی بشه و مبدأ و مقصد یکی باشن
* شرط وجود در گراف بدون جهت:

مدار اویلری:

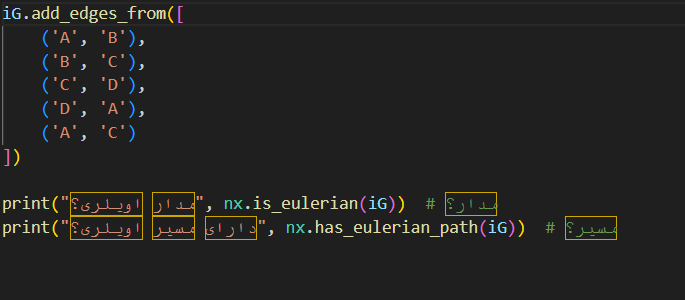
تمام رئوس باید درجه زوج داشته باشن

مثلاً: 2، 4، 6 ...

مسیر اویلری (اما نه مدار):

دقیقاً دو رأس درجه فرد داشته باشن

بقیه درجه زوج باشن.



* شرط وجود در گراف جهت‌دار:

مدار اویلری:

برای هر رأس: درجه ورود = درجه خروج

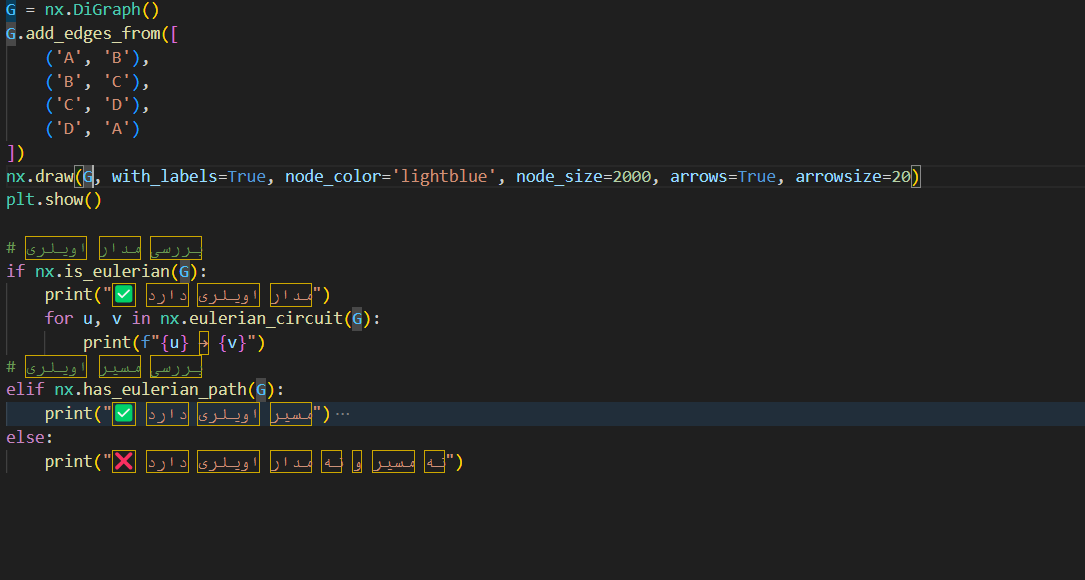
مسیر اویلری:

دقیقاً یک رأس باشه که:

درجه خروج = درجه ورود + 1 (شروع مسیر)

و دقیقاً یک رأس که:

درجه ورود = درجه خروج + 1 (پایان مسیر)



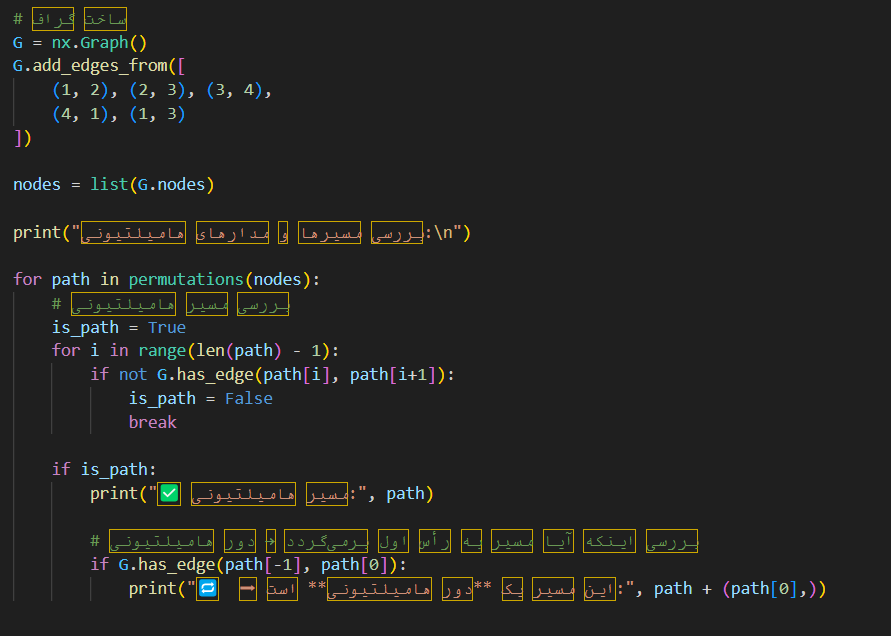
* دور هامیلتیونی (Hamiltonian Cycle)

تعریف: دور هامیلتیونی (یا چرخه هامیلتیونی)، مسیر بسته‌ای است که:

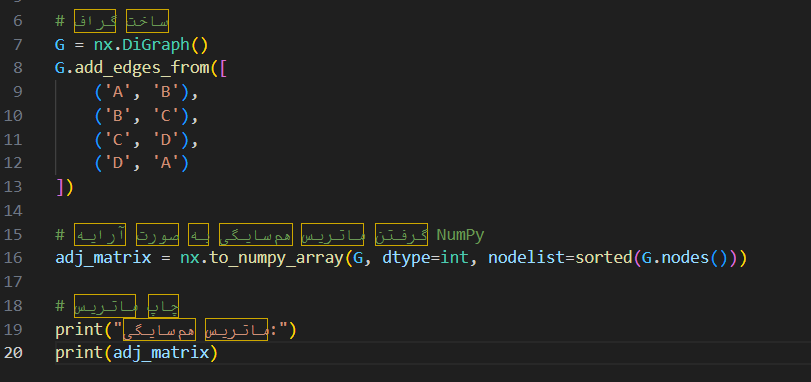
از هر رأس دقیقاً یک‌بار عبور می‌کند،و در نهایت به رأس شروع برمی‌گردد.

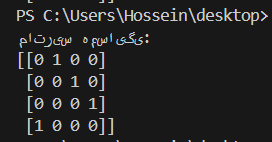
* مسیر هامیلتیونی (Hamiltonian Path)

تعریف: مسیر هامیلتیونی، مسیری در یک گراف است که از هر رأس دقیقاً یک‌بار عبور کند (بدون تکرار رأس‌ها)، اما لازم نیست به رأس شروع برگردد.



* برای گراف‌ها، ماتریس هم‌سایگی (Adjacency Matrix) یک نمایش ماتریسی از ارتباط بین گره‌هاست.

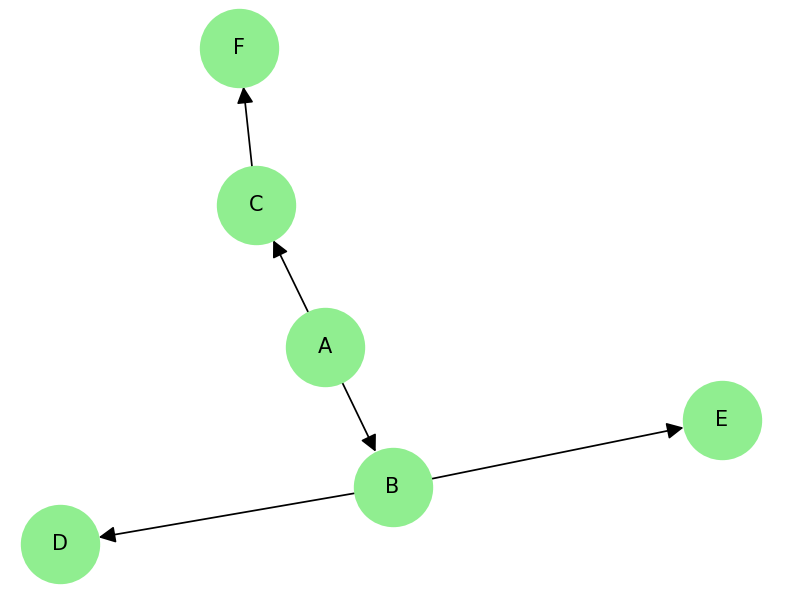




* Directed Tree (درخت جهت‌دار): درخت جهت‌دار یک نوع گراف جهت‌دار است که ویژگی‌های زیر را دارد:

1. یک گره ریشه (Root) دارد که هیچ یال ورودی ندارد.
2. از ریشه می‌توان با دنبال کردن یال‌ها به تمام گره‌های دیگر رسید.a
3. گراف بدون دور (cycle) است.
4. هر گره (غیر از ریشه) دقیقاً یک یال ورودی دارد.

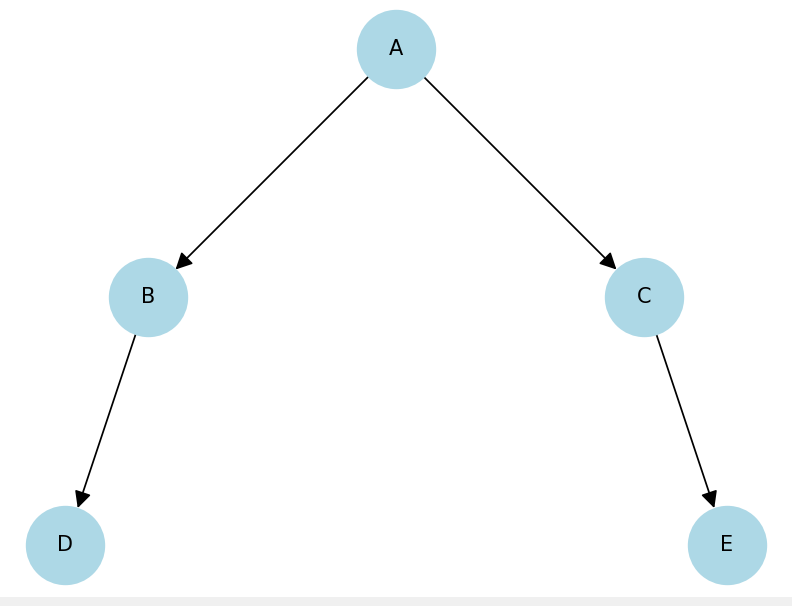




* Binary Tree (درخت دودویی) چیست؟ درخت دودویی ساختاری درختی است که:

1. هر گره می‌تواند حداکثر دو فرزند داشته باشد:
2. یک فرزند چپ (Left Child)
3. یک فرزند راست (Right Child)

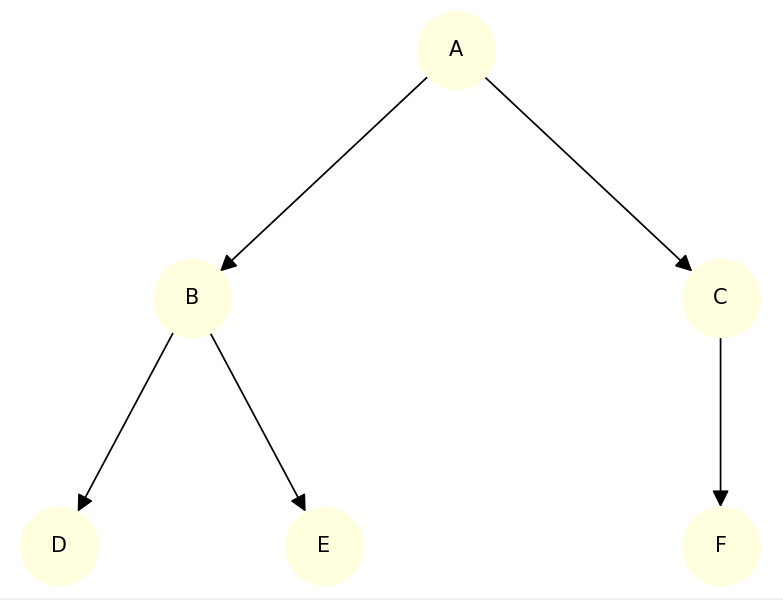




* درخت مرتب (Ordered Tree): نوعی درخت ریشه‌دار هست که در اون

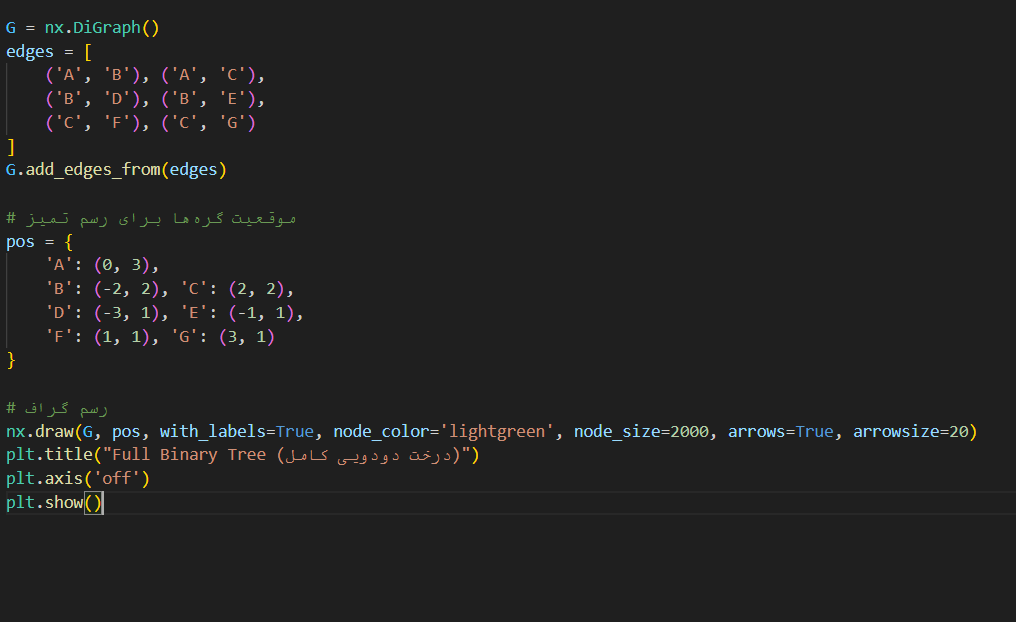
1. هر گره می‌تونه چند فرزند داشته باشه
2. ترتیب فرزندان اهمیت داره (یعنی فرق هست بین فرزندان [B, C] و [C, B])

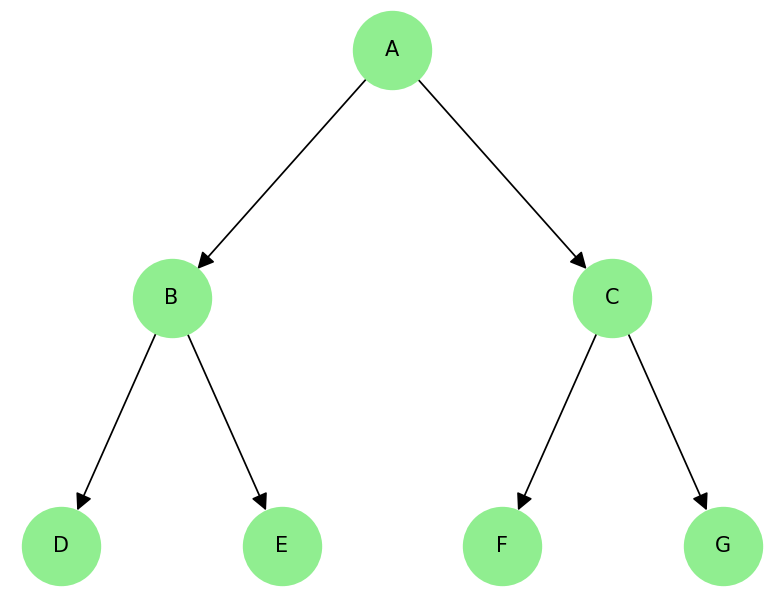




* درخت دودویی کامل (Full Binary Tree) نوعی درخت دودوییه که در اون:

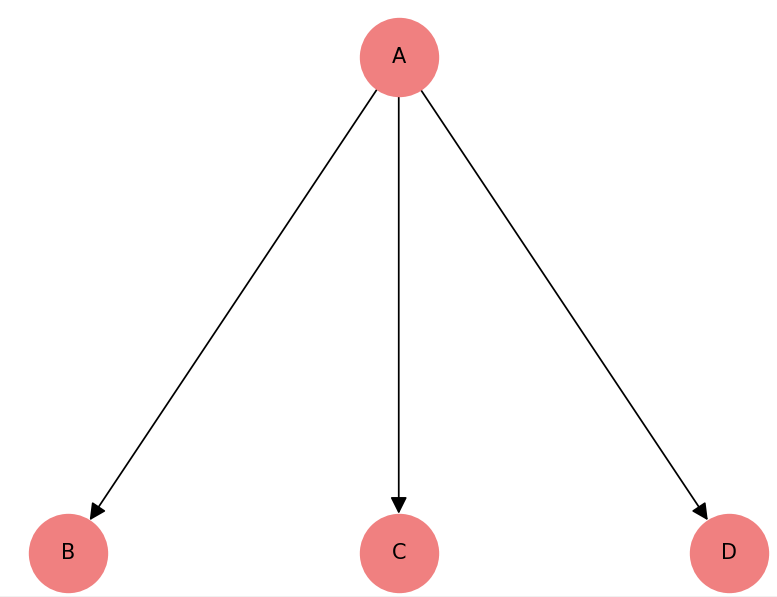
1. هر گره یا دقیقاً دو فرزند داره یا هیچ فرزندی نداره.
2. یعنی هیچ گره‌ای فقط یک فرزند نداره.





* Positional Tree یا درخت موقعیتی نوعی درخت مرتب (Ordered Tree) هست که:

علاوه بر ترتیب بین فرزندان، موقعیت دقیق هر فرزند در یک مکان خاص (Position) تعریف شده.

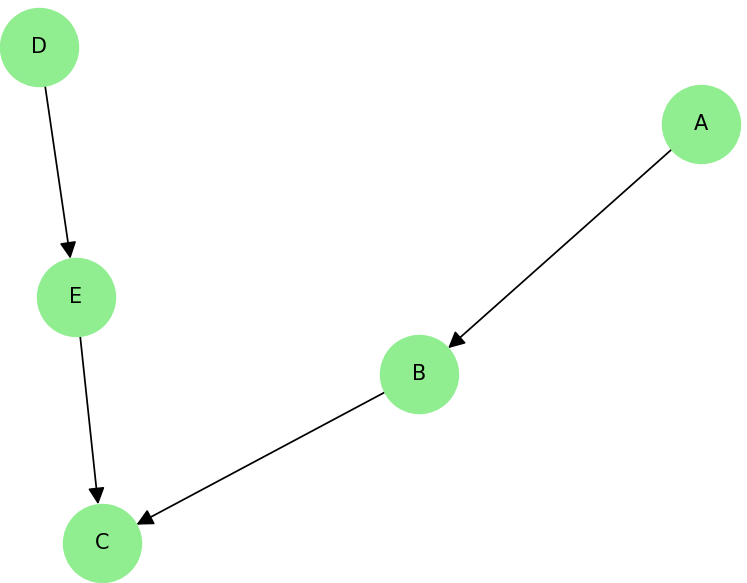




* یک گراف جهت‌دار (Directed Graph) وقتی متصل ضعیف (Weakly Connected) محسوب می‌شه که:

1. اگر جهت یال‌ها را نادیده بگیریم، گراف بدون جهت حاصل، متصل باشد.
2. یعنی شاید از A به B مسیر مستقیم نباشه، اما اگه جهت‌ها رو حذف کنیم، همه گره‌ها به هم وصل باشن.





* دو گراف جهت‌دار (DiGraph) رو یک‌ریخت (isomorphic) می‌گیم اگر:

بتونیم برچسب گره‌ها رو بازچینش کنیم طوری که ساختار یال‌ها دقیقاً یکسان باقی بمونه (با جهت‌ها).

* **شرط‌ها:**

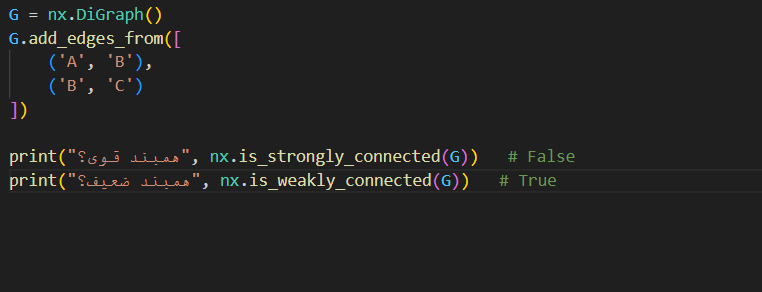
1. دو گراف جهت‌دار G و H یک‌ریخت هستن اگر:
2. تعداد گره‌ها برابر باشه
3. جهت و اتصال یال‌ها یکی باشه
4. فقط اسم گره‌ها فرق کنه



خروجی : true و false

گراف یک ریخت غیر جهت دار نیز همین طور است .

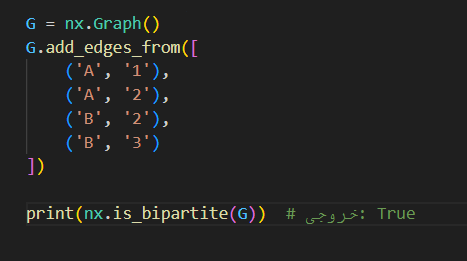
* گراف همبند قوی (Strongly Connected Graph) چیست؟ گرافی که بتوان از هر راس آن به راس های دیگر رفت و دسرسی داشت برعکس همبند ضعیف .



* گراف جهت دار متصل یک طرفه:همان گراف همبند ضعیف میتواند باشد یعنی از یک راس میتوان به راس دیگه رفت اما برگشتی ندارد فقط از یک رفت وصل است .
* گراف دوبخشی یعنی:

می‌تونی همهٔ رأس‌ها (نقطه‌ها) رو به دو دسته تقسیم کنی

طوری که هیچ یالی (خطی) بین دو رأس از یه دسته نباشه.



* هدف الگوریتم کراسکال:

1. پیدا کردن یک درخت پوشای کمینه (MST)

→ یعنی کمترین مجموعه‌ی یال‌ها که همه رأس‌ها رو وصل کنه و دور نداشته باشه.

* مراحل اجرای الگوریتم کراسکال:

مرتب کردن یال‌ها بر اساس وزن:

A-B → 1

B-C → 2

A-C → 3

ساخت MST (درخت پوشا):

مرحله 1:

A-B رو انتخاب می‌کنیم → وزن = 1

(A و B متصل شدن)

مرحله 2:

B-C رو انتخاب می‌کنیم → وزن = 2

(C هم به جمع وصل می‌شه)

مرحله 3:

A-C وزنش 3 هست، ولی اگر انتخابش کنیم، یه دور ایجاد می‌شه پس رد می‌کنیم.

نتیجه نهایی:

درخت پوشای کمینه شامل یال‌های:

A-B (وزن 1)

B-C (وزن 2)

مجموع وزن = 1 + 2 = 3

