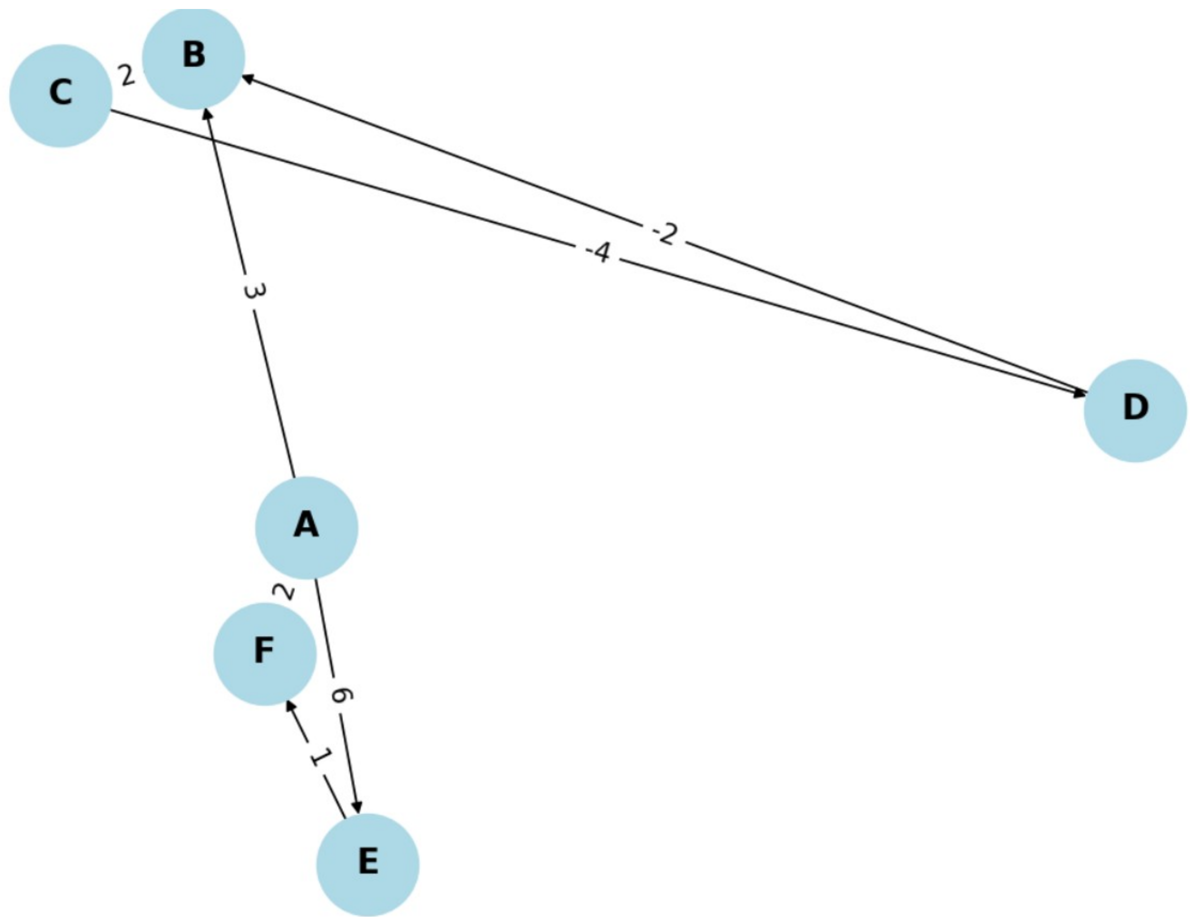
سیاه چاله

شما وارد دنیای پیچیده‌ای از گراف‌ها شده‌اید، جایی که هر گوشه و هر یال، معنای خاصی دارد. اما این دنیا یک تهدید پنهانی دارد که در اعماق آن منتظر شماست... **سیاه چاله**!

این سیاه چاله که در حقیقت **چرخه منفی** است، به طور مخفیانه در گراف شما حضور دارد. اگر شما در این مسیر گمراه شوید، ممکن است برای همیشه در این چرخه منفی گرفتار شوید و هیچ راه فراری پیدا نکنید!

مهم: تنها یک سیاه چاله در این گراف وجود دارد، و شما باید با استفاده از هوش خود این مسیر خطرناک را پیدا کنید و از افتادن در آن جلوگیری کنید. برای این کار، از الگوریتم‌های مناسب استفاده کرده و **چرخه منفی** (همان سیاه چاله) را شناسایی کنید.

چالش: شما باید با دقت و احتیاط قدم بردارید، چون تنها یک اشتباه کوچک می‌تواند شما را در این دنیای بی‌پایان و منفی فرو ببرد. آیا می‌توانید **سیاه چاله** را پیدا کرده و از آن نجات یابید؟



الگوریتمی که استفاده کردید تا این سیاه چاله را پیدا کنید را توصیف کنید و مراحل انجام آن الگوریتم را نیز بنویسید

الگوریتم Bellman-Ford برای پیدا کردن کوتاه‌ترین مسیر از یک مبدأ به همه‌ی رأس‌ها استفاده می‌شود و می‌تواند وجود دور منفی را نیز تشخیص دهد. پس از انجام n-1 بار به‌روزرسانی وزن‌ها (relaxation)، اگر هنوز هم بتوان مسیری را با وزن کمتر پیدا کرد، به این معناست که یک دور منفی در گراف وجود دارد. یعنی اگر بعد از n−1 بار بررسی یال‌ها، هنوز هم مسیری با هزینه کمتر پیدا شود، می‌توانیم بی‌نهایت بار از آن چرخه استفاده کنیم تا هزینه مسیر را کمتر و کمتر کنیم، در نتیجه یک چرخه منفی وجود دارد.

برای گراف داده شده از گره A شروع کرده و V-1 = 6 بار هر راس را Relax می‌کنیم.
تعدادی از مراحل آپدیت کردن:

Dist = {0, INF, INF, INF, INF, INF}

Dist[1] > Dist[0] + 3, Dist[4] > Dist[0] + 6

Dist = {0, 3, INF, INF, 6, INF}

Dist[5] > Dist[4] + 1, Dist[2] > Dist[1] + 2

Dist = {0, 3, 5, INF, 6, 7}

Dist[3] > Dist[2] + (-4)

Dist = {0, 3, 5, 1, 6, 7}

Dist[1] > Dist[3] + (-2)

Dist = {0, -1, 5, 1, 6, 7}

Dist[1] > Dist[3] + (-2)

Dist = {0, -3, 5, 1, 6, 7}

پس ۶ مرحله، برای گره B، تا بی‌نهایت مسیر کوتاه‌تر پیدا می‌شود پس گراف دارای دور منفی است.
زمانی که تشخیص می‌دهیم `Dist[v]` در تکرار nام دوباره آپدیت شده، آن یال نشان‌دهنده عضوی از چرخه‌ی منفی است. از آن گره با دنبال‌کردن آرایه‌ی `parent` می‌توانیم به عقب برویم تا چرخه کامل را پیدا کنیم.

B → C → D → B

1 → 2 → 3 → 1