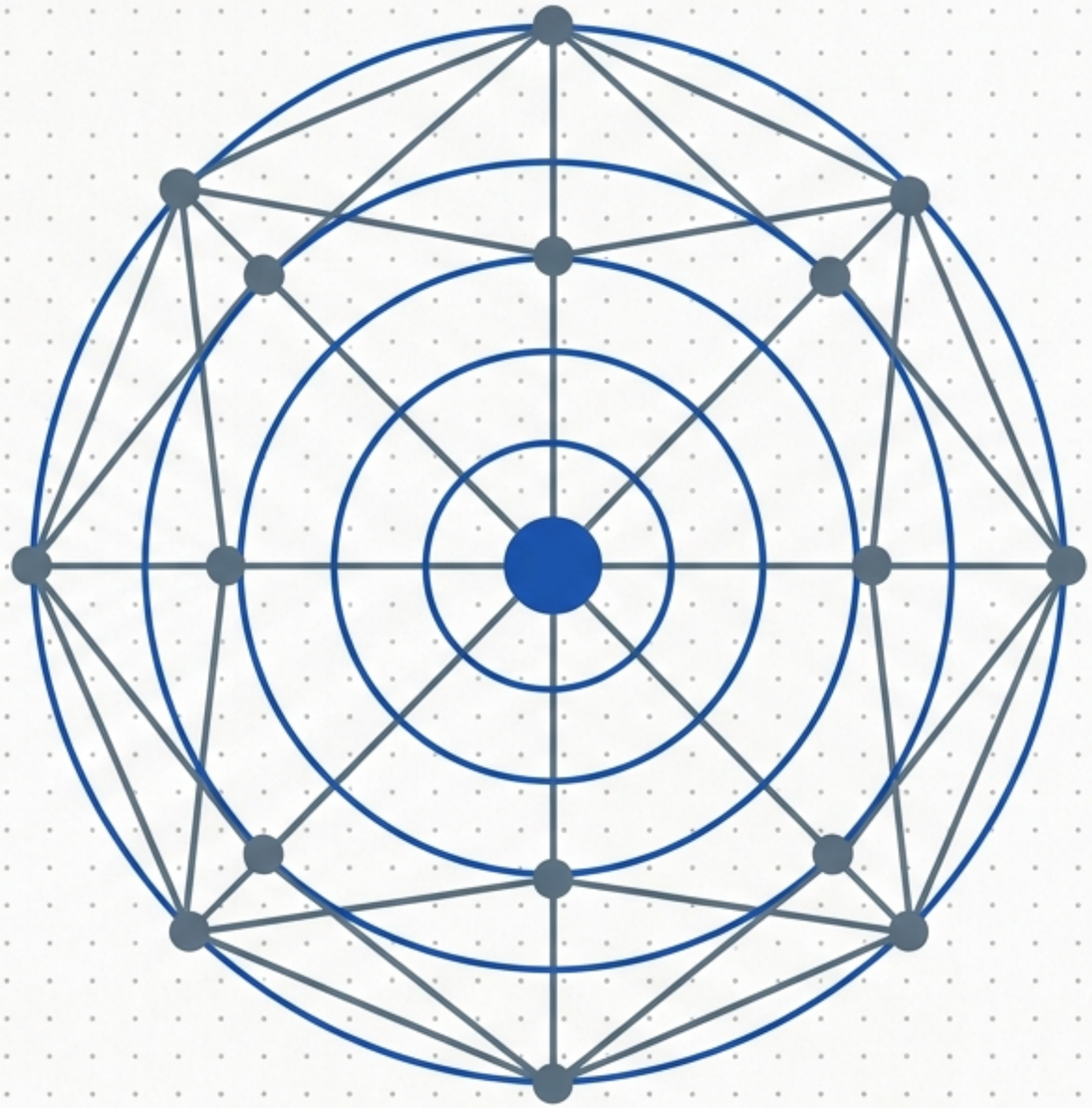
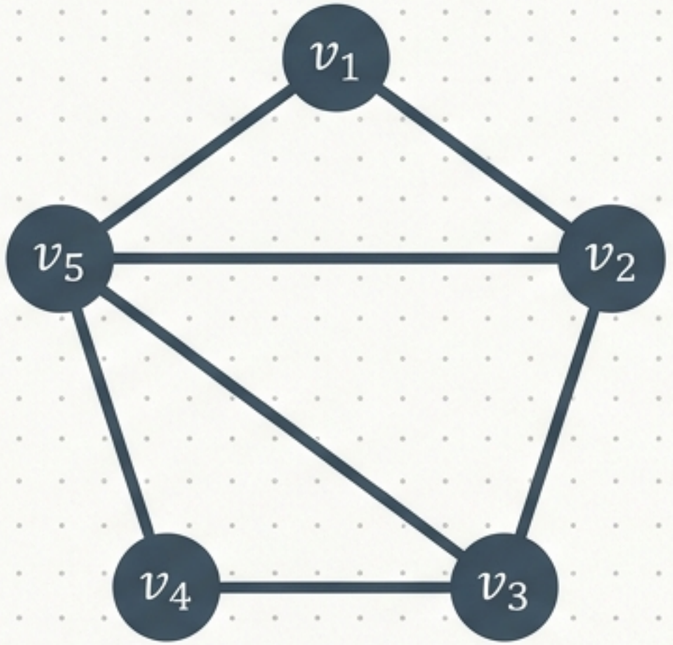
راز «رئوس زیبا»

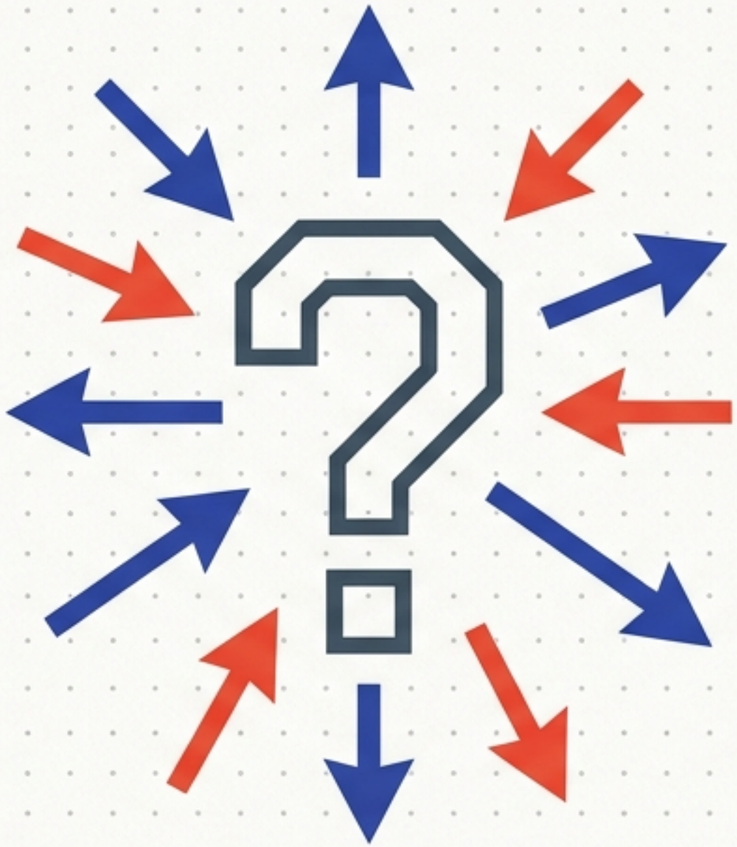
کالبدشکافی مسیرهای متناوب و گراف‌های دوبخشی در طراحی الگوریتم

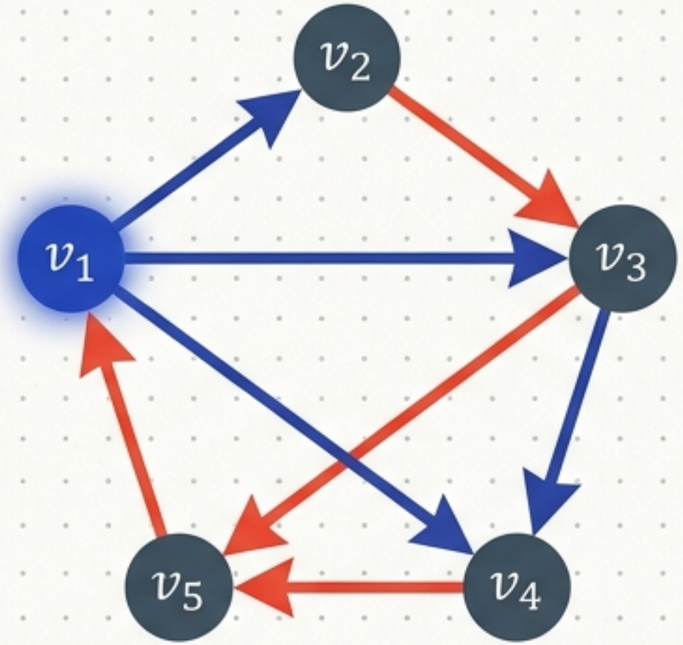
یک سفر هیجان‌انگیز از معمای پیچیده‌ی گراف
به یک راه‌حل ظریف با پیچیدگی زمانی O(V+E)



NotebookLM







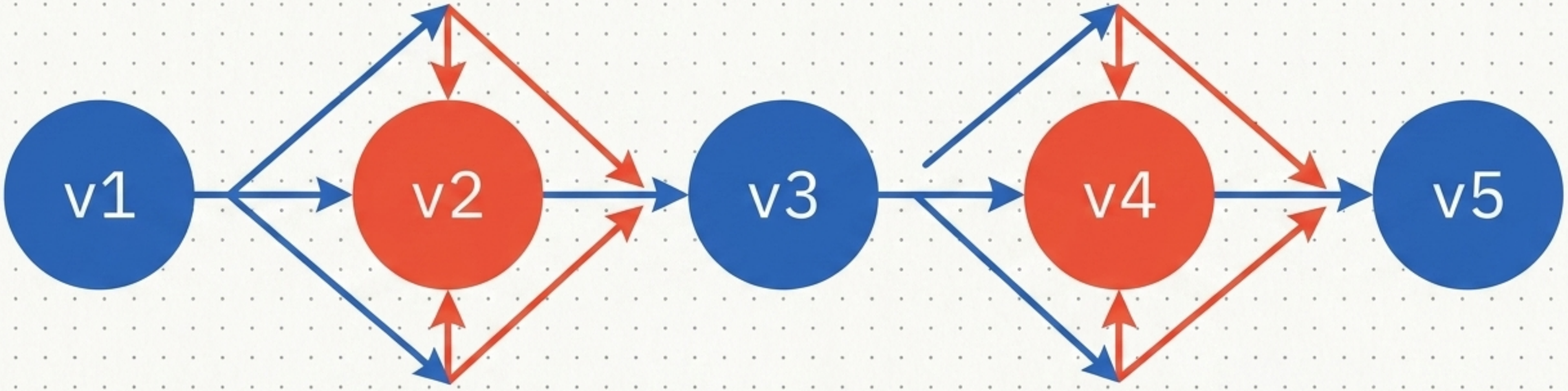
داده‌های ورودی: یک گراف ساده $G(V, E)$ بدون طوقه (Self-loop) یا یال چندگانه.

ماموریت اصلی: جهت‌دهی به تمام یال‌ها برای بیشینه‌سازی تعداد «رئوس زیبا».

تعریف شرط: رأس v زیباست اگر تمام مسیرهای شروع‌شده از آن، یک «مسیر متناوب» (Alternating Path) باشند.

NotebookLM

کالبدشکافی ساختار «مسیر متناوب»



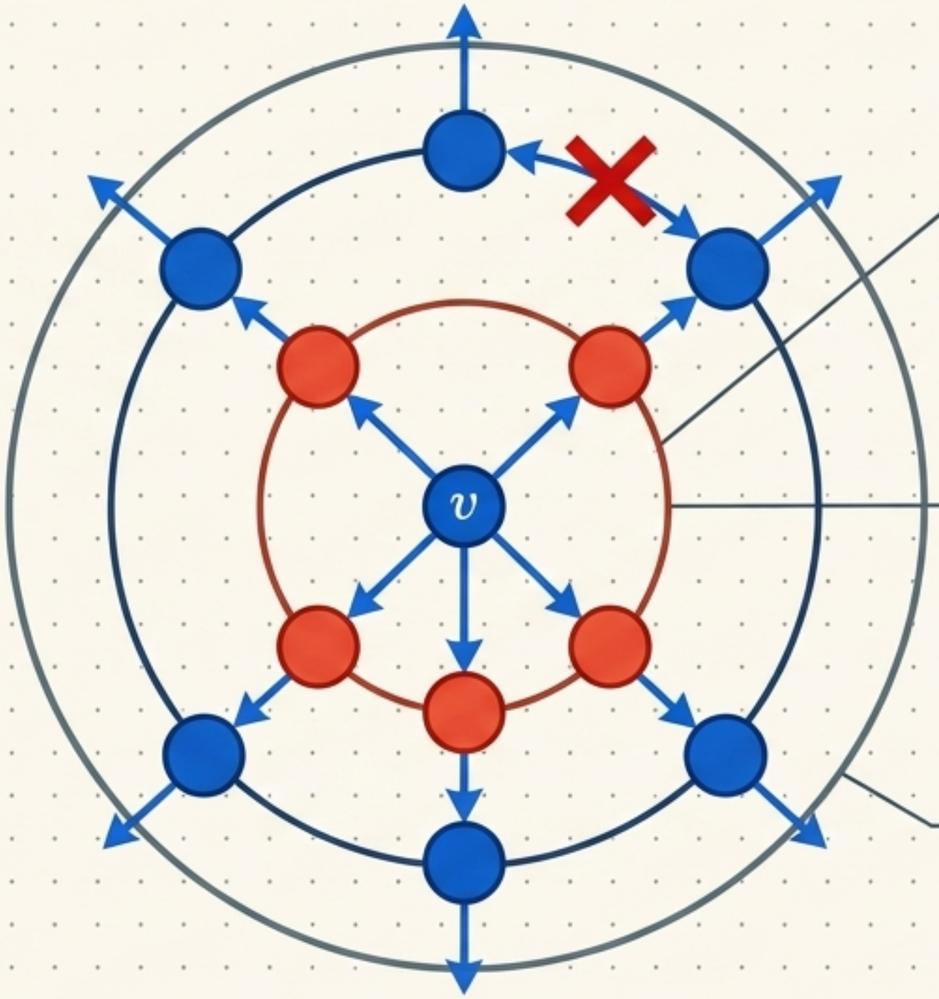
گره‌های فرد: نقش «چشمه» (Source).
یال‌ها فقط از آن‌ها خارج می‌شوند.

گره‌های زوج: نقش «چاه» (Sink).
یال‌ها فقط به آن‌ها وارد می‌شوند.

مسیر متناوب در واقع یک نوسان و تپش دائمی بین مفهومِ چشمه و چاه است.

NotebookLM

یک «رأس زیبا» دقیقاً چه محدودیت‌هایی خلق می‌کند؟



باید «چشمه مطلق» باشد (فقط خروجی).

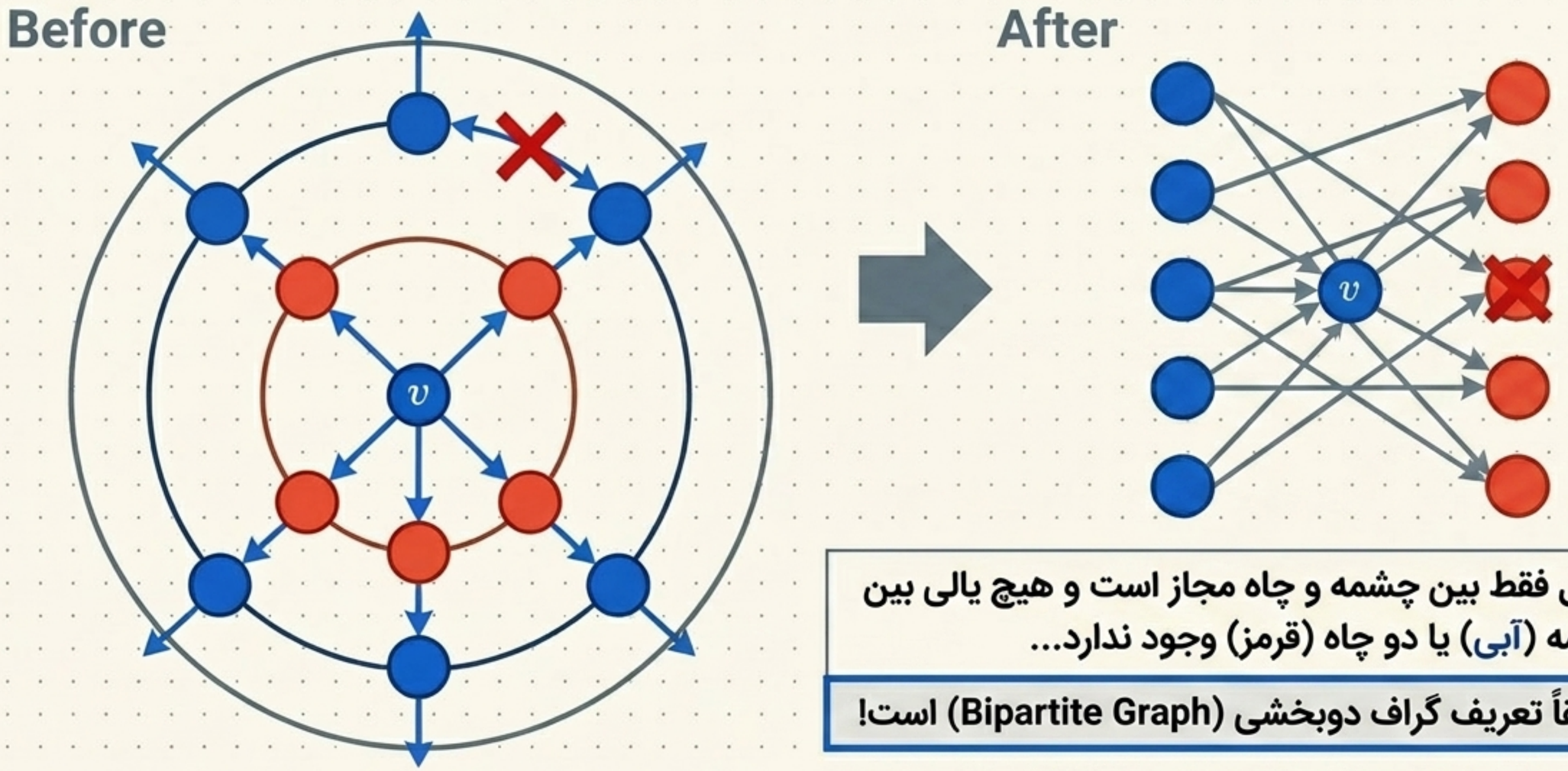
تمام همسایه‌ها باید «چاه مطلق» باشند (اگر یال خروجی داشته باشند، تناوب مسیر نقض می‌شود).

همسایه‌های همسایه‌ها مجدداً «چشمه مطلق» می‌شوند.

نتیجه: یک رأس زیبا، کل مؤلفه هم‌بند را مجبور می‌کند تا نقش‌های متناوب به خود بگیرند.

NotebookLM

لحظه «آها!»: راز نهفته در دل مسئله



چون یال فقط بین چشمه و چاه مجاز است و هیچ یالی بین دو چشمه (آبی) یا دو چاه (قرمز) وجود ندارد...

این دقیقاً تعریف گراف دوبخشی (Bipartite Graph) است!

قانون طلایی: اگر یک مؤلفه از گراف قابلیت رنگ‌آمیزی با دو رنگ را داشته باشد، پتانسیل خلق رئوس زیبا را دارد.

NotebookLM

استراتژی تصمیم‌گیری (بهینه‌سازی هر مؤلفه)

آیا مؤلفه گراف دوبخشی است؟
(بدون دور فرد)

خیر

تناقض (وجود دور فرد): در صورت برخورد دو گره هم‌رنگ، هیچ رأسی در آن در آن بخش هم بند نمیتواند نمی‌تواند زیبا باشد.

(سهم = ۰)

بله

Set A | Set B

حالت ۱: جهت‌یابی A به B
(تمام رئوس A زیبا می‌شوند).

حالت ۲: جهت‌یابی B به A
(تمام رئوس B زیبا می‌شوند).

برای بیشینه‌سازی، مقدار $\max(|A|, |B|)$ را جواب نهایی اضافه می‌کنیم.

NotebookLM

الگوریتم نهایی: گام‌به‌گام تا راه‌حل

Step 1: پیمایش (Traversal)

استفاده از الگوریتم BFS یا DFS برای کشف و پیمایش کن مؤلفه‌های هم‌بندِ بررسی‌نشده.

Step 2: رنگ‌آمیزی (Coloring)

اختصاص دو رنگ (مثلاً ۰ و ۱)) به لایه‌های متناوب در حین حین پیمایش.

Step 3: اعتبارسنجی (Validation)

بررسی وجود «دور فرد». اگر یک گره به همسایه‌ای با رنگ مشابه متصل باشد، کل آن مؤلفه رد می‌شود.

Step 4: محاسبه (Calculation)

در صورت معتبر بودن، تعداد گره‌های هر رنگ شمارش شده و عدد بزرگتر max(Count_0, Count_1) جواب نهایی اضافه می‌گردد. می‌گردد.

NotebookLM

چالش مهندسی: انتخاب ساختمان داده مناسب

	ماتریس مجاورت (Adjacency Matrix)	لیست مجاورت (Adjacency List)
حافظه مصرفی	✗ O(V^2)	✓ O(V+E)
یافتن همسایه‌ها (در BFS)	✗ بررسی V خانه برای هر گره	✓ پیمایش مستقیم deg(v) همسایه واقعی
پیچیدگی کل اجرای الگوریتم	گلوگاه در ساخت ماتریس و خواندن → O(V^2)	سرعت ایده‌آل → O(V+E)

برای جلوگیری از ایجاد گلوگاه (Bottleneck) در گراف‌های خلوت (Sparse Graphs)، استفاده از لیست مجاورت (لینک‌لیست) از همان لحظه خواندن ورودی الزامی است.

NotebookLM

پیاده‌سازی معماری (زبان Java)

```java
ArrayList<ArrayList<Integer>> adj;
Queue<Integer> q = new LinkedList<>();

int[] color = new int[V + 1];
Arrays.fill(color, -1);

// BFS loop...
if (color[neighbor] == -1) {
    color[neighbor] = 1 - color[curr];
    q.add(neighbor);
} else if (color[neighbor] == color[curr]) {
    return -1; // Odd cycle found
}
```

ساختار داده ایده‌آل: ذخیره شبکه با استفاده از لیست مجاورت برای دسترسی سریع.

پیمایش سطح‌به‌سطح: کنترل منطق چشمه/چاه با رنگ‌آمیزی ۰ و ۱ در صف BFS.

شناسایی دور فرد: بازگشت مقدار 1- در صورت برخورد با گره هم‌رنگ (تناقض دوبخشی).

NotebookLM

تحلیل پیچیدگی نهایی (Performance Dashboard)

O(V + E)

زمان (Time Complexity)

پیمایش ورودی و اکتشاف گره‌ها/یال‌ها در لیست مجاورت، دقیقاً یک بار برای هر مؤلفه انجام می‌شود.

O(V + E)

حافظه (Space Complexity)

تخصیص حافظه تنها متناسب با یال‌های واقعی (لیست مجاورت)، آرایه رنگ‌ها و صف BFS.

Maximum Efficiency Achieved: تبدیل موفقیت‌آمیز یک معمای زمان‌بر یا. به یک ساختار خطی و قطعی.

NotebookLM

دستاوردهای کلیدی این معماری

اهمیت ابزار

زیباترین الگوریتم O(V+E) نیز اگر در دام ساختمان داده غلط (ماتریس مجاورت با زمان O(V^2)بیفتد، کارایی خود را از دست می‌دهد. انتخاب لیست مجاورت کلید مقیاس‌پذیری بود.

استراتژی بهینه‌سازی

بیشینه‌سازی در این شبکه پیچیده، با یک شمارش ساده و انتخاب بزرگترین بخش رنگ‌آمیزی‌شده (۲-Color Partition) محقق شد.

راز تعاریف

«مسیر متناوب» تنها یک نقاب ظریف برای پنهان کردن مفهوم کلاسیک گراف دوبخشی بود. کشف الگوها نیمی از راه‌حل است.

NotebookLM