

گزارش سؤال ۳

ابتدا یال‌ها را در ورودی می‌گیریم و گراف را به دو صورت لیست یال‌ها (edges) و لیست‌های مجاورت (adjacency_dict) می‌سازیم (می‌توانستیم لیست یال هم نسازیم ولی در این حالت کد کمی تمیزتر است. زمان و حافظه هم با این کار حداکثر دوبرابر می‌شود و دوبرابر شدن اردر جواب را تغییر نمی‌دهد)

```
5 V, E = map(int, input().split())
6 edges = []
7 for i in range(E):
8     v1, v2 = map(int, input().split())
9     edges.append((v1, v2))
10
11 adjacency_dict = defaultdict(list)
12 for edge in edges:
13     adjacency_dict[edge[0]].append(edge[1])
14     adjacency_dict[edge[1]].append(edge[0])
```

طبق راهی که در [این لینک](#) پیشنهاد شده مراحل زیر را انجام می‌دهیم:

تا وقتی که مجموعه vertex‌هایمان همه‌ی یال‌ها را پوشش نمی‌دهد:

1. رأس v که بیشترین درجه را دارد انتخاب می‌کنیم
2. از راس‌هایی که به v یال دارد آن را که درجه‌اش از همه بیشتر است به نام u انتخاب می‌کنیم (هر دوی u و v تا الان در مجموعه vertex‌هایمان اضافه نشده بوده‌اند)
3. u و v را به مجموعه vertex‌هایمان اضافه می‌کنیم (شرطی که خودم اضافه کردم این بود که اگر u درجه‌اش در گراف ۱ بود دیگر u را اضافه نکند چون معنیش این است که فقط به v یال دارد و بنابراین اضافه کردن v به تنهایی یال بینشان را cover می‌کند.)

4. یال‌هایی که یک سرشان u یا v است از گراف حذف می‌کنیم.

5. Go to step 1

و نهایتاً هم vertex cover بدست آمده را چاپ کند.

```
16 vert_cover = set()
17 while len(edges) > 0:
18     v = max(adjacency_dict.items(), key=lambda x: len(x[1]))[0]
19     # u = random.choice(adjacency_dict[v])
20     u = max({key: adjacency_dict[key] for key in adjacency_dict[v]}.items(), key=lambda x: len(x[1]))[0]
21     vert_cover.add(v)
22     if len(adjacency_dict[u]) > 1:
23         vert_cover.add(u)
24     for key, val in adjacency_dict.items():
25         adjacency_dict[key] = [vertex for vertex in val if not vertex in [u, v]]
26     del adjacency_dict[v]
27     del adjacency_dict[u]
28     edges = list(filter(lambda x: (not x[0] in [u, v]) and (not x[1] in [u, v]), edges))
29
30 print(" ".join(list(map(str, list(vert_cover)))))
31
```

علت approximate-2 بودنش این است که الگوریتم ما دارد بصورت حریصانه maximal matching در گراف پیدا می‌کند. طبق تعریف maximal matching، هر یالی در گراف به یکی از رئوس موجود در maximal matching وصل است (چون اگر یالی اینطور نباشد می‌تواند به maximal matching اضافه شود و در آن صورت دیگر maximal نیست). بنابراین vertex cover کمینه حداقل به اندازه نصف تعداد رئوس دو سر یال‌های maximal matching و حداکثر به اندازه برابر با آن عضو دارد ([منبع](#))

مرتبه زمانی آن هم از نظر شرط loop و عملیات حذف مثل approximate vertex cover معمولی است که $O(V + E)$ است.