الگوریتم min-cut

توضیح الگوریتم :

در این پروژه میخواهیم با حذف کردن کمترین یال ها از گراف داده شده به عنوان ورودی ، گراف را به 2 زیر گراف جدا از هم تقسیم کنیم . برای این کار به این ترتیب عمل میکنیم :

ابتدا یک یال رندوم در نظر می گیریم سپس رئوس دو سر یال را با هم ادغام می کنیم دو سر یال را یک راس در نظر میگیریم این کار را ادامه می دهیم تا در نهایت دو راس باقی بماند . یال های میان دو راس باقی مانده به عنوان یال هایی که باید حذف شوند معرفی می شوند . باید توجه کرد که چون الگوریتم برش مینمم از نوع الگوریتم های مونت کارلو هستند بنابراین ممکن است که خروجی نهایی مینیمم نباشد .

```
1)  Initialize contracted graph CG as copy of original graph
2)  While there are more than 2 vertices.
      a) Pick a random edge (u, v) in the contracted graph.
      b) Merge (or contract) u and v into a single vertex (update
         the contracted graph).
      c) Remove self-loops
3) Return cut represented by two vertices.
```

پیچیدگی زمانی :

$O(E) = O(V^2)$

توضیح کد :

:FindMinimum

در این تابع ، تابع اصلی fastMinCut چند بار فراخوانی میشود تا کمترین مقدار آن به دست آید.

```
edge = min(edge, fastMinCut(graph))
```

:FastMinCut

```
procedure fastmincut(G = (V, E)):
if |V| ≤ 6:
    return mincut(V)
else:
    t ← ⌈1 + |V|/√2⌉
    G1 ← contract(G, t)
    G2 ← contract(G, t)
    return min {fastmincut(G1), fastmincut(G2)}
```

:MinCut

الگوریتم اصلی (همان karger's algorithm) است که یک یال رندوم انتخاب میکندو راس های دو سر آن را ادغام میکند.

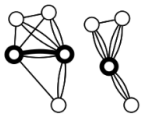
:EdgeCount & VertexCount

این دوتابع تعداد راس ها و یالها را برمیگردانند.

:Contract

این تابع دو راس ابتدا و انتهای یک یال را ادغام میکند.

عملکرد بنیادی الگوریتم کارگر شکلی از *تلفیق یالی* است. نتیجهٔ تلفیق یال $e = \{u, v\}$ رأس تازهٔ uv خواهد بود. هر یال $\{w, u\}$ یا $\{w, v\}$ برای $w \notin \{u, v\}$ در دو انتهای یال تلفیقی با یک یال $\{w, uv\}$ به رأس تازه جایگزین می‌شود. سرانجام، رئوس تلفیقی u و v با همهٔ یال‌های واقع بر آن حذف می‌شوند. به ویژه گراف حاصل هیچ حلقه‌ای را در بر ندارد. نتیجهٔ تلفیق یال e با G/e نشان داده می‌شود.



الگوریتم تلفیق مرتباً یال‌های تصادفی را در گراف تلفیق می‌کند تا زمانی که فقط دو رأس باقی بماند که آنگاه فقط یک برش یکتا وجود دارد.

اجرای موفق الگوریتم کارگر روی یک گراف ۱۰ رأسی. تعداد یال‌های برشی کمینه ۳ است.

```
procedure contract (G = (V, E)) :
while |V| > 2
    choose e ∈ E uniformly at random
    G ← G/e
return the only cut in G
```

ورودی :

```
graph = Graph([[1, 2, 3, 4], [2, 1, 3, 4], [3, 1, 2, 4], [4, 2, 3, 1]])
```

خروجی :

```
Run:  min-cut
C:\Users\Ali\.virtualenvs\mb-BnrY8yko\Scripts\python.exe D:/prozhe/min-cut/min-cut.py
minimum cut(s) = 3

Process finished with exit code 0

4: Run  6: TODO  DB Execution Console  9: Version Control  Python Console  Terminal
```