آزمایشگاه ریاضی (آموزش Sage)

درس چهارم:
نظریه گراف

مدرس: میثم مدنی madani@mehr.sharif.ir
دانشکده علوم ریاضی- دانشگاه صنعتی شریف

ترم اول سال ۱۳۹۱

نسخه ابتدایی (لطفا نظرات خودت و ایرادات فایل را ایمیل کنید)

آشنایی

```
G=Graph({0:[1,2,3], 2:[4]})
G=Graph({0:{1:"x",2:"z",3:"a"}, 2:{5:"out"}})
```

تعریف دو گراف یکی به صورت ساده و دیگری لیبل دار

x,z,a,out

```
G = Graph([(1,3,"Label"),(3,8,"Or"),(5,2)])
```

گراف ماتریس ها

ماتریس مجاورت (ماتریس باید مجاورت باشد!)

```
M=Matrix([[0,1,1,0],[1,0,1,1],[1,1,0,1],[0,1,1,0]])
G = Graph(M)
G.show()
```

ماتریس وقوع

```
N=Matrix([[-1,-1,0,0,0],[1,0,-1,-1,0],[0,1,1,0,-1],
[0,0,0,1,1]])
H=Graph(N, format='incidence_matrix')
H.show()
I = g.incidence_matrix()
```

اختیارات در گرافها

- G.show(edge_labels=True)
- Graph(M,(multiedges = True))
- G= copy(H)
- g.plot(graph_border=True).save('graph.pdf')
- g.edges()
- g.vertices()
- g.plot(layout='circular')
- g.plot(layout='tree')
- g.is_planar(set_pos=True);g.plot()
- g.plot3d()

استفاده از گرافهای پیش فرض

خود sage گرافهای بسیاری را به صورت از پیش تعریف شده در بر دارد.
برای اینکار کافیست از دستوری مثل

graphs.PetersenGraph()

استفاده کنیم. که البته می توانید به جای PetersenGraph اسم گراف مورد نظر را بیاورید.

اگر اسمش را نمی دانید کافیست بعد از .graphs یک Tab بزنید. از لیست یکی را انتخاب کنید.

(معمولا کلیدهای میانبر در internet explorer کار نمی کنند)

تبدیلات

G.to_directed()

G.to_undirected()

G.sparse6_string()

G.graph6_string()

ضرب ها

- G.strong_product(H)
- G.tensor_product(H)
- G.categorical_product(H)
- G.disjunctive_product(H)
- G.lexicographic_product(H)
- G.cartesian_product(H)

سوالات بله خیر

- G.is_tree()
- G.is_forest()
- G.is_gallai_tree()
- G.is_interval()
- G.is_regular()
- G.is_chordal()
- G.is_eulerian()
- G.is_hamiltonian()
- G.is_interval()
- G.is_independent_set([vertices])
- G.is_overfull()
- G.is_regular(k)

ناورداهای رایج

- G.diameter()
- G.average_distance()
- G.edge_disjoint_spanning_trees(k)
- G.girth()
- G.size()
- G.order()
- G.radius()

رنگ آمیزی گرافها

- G.chromatic_polynomial()
- G.chromatic_number(algorithm="DLX")

You can change DLX (dancing links) to CP (chromatic polynomial coecients) or MILP (mixed integer linear program)

- G.coloring(algorithm="DLX")

You can change DLX to MILP

- G.is_perfect(certificate=False)

مسطح بودن

- G.is_planar()
- G.is_circular_planar()
- G.is_drawn_free_of_edge_crossings()
- G.layout_planar(test=True, set_embedding=True)
- G.set_planar_positions()

جستجوها و کوتاهترین مسیر

- list(G.depth_first_search([vertices], distance=4)
- list(G.breadth_first_search([vertices])
- dist,pred = graph.shortest_path_all_pairs(by_weight)

Choice of algorithms: BFS or Floyd-Warshall-Python

- G.shortest_path_length(v_1,v_2, by_weight=True)
- G.shortest_path_lengths(v_1)
- G.shortest_path(v_1,v_2)

درخت فراگیر

- G.steiner_tree(g.vertices()[:10])
- G.spanning_trees_count()
- G.edge_disjoint_spanning_trees(2, root vertex)
- G.min_spanning_tree(weight_function=somefunction, algorithm='Kruskal',starting_vertex=3)

Kruskal can be change to Prim fringe, Prim edge, or NetworkX

جبر گرافها

- Matrices
- G.kirchhoff_matrix()
- G.laplacian_matrix()
- G.weighted_adjacency_matrix()
- G.adjacency_matrix()
- G.incidence_matrix()

- G.characteristic_polynomial()
- G.cycle_basis()
- G.spectrum()
- G.eigenspaces(laplacian=True)
- G.eigenvectors(laplacian=True)

اتومورفیسم ها و ایزومورفیسم ها

- G.automorphism_group()
- G.is_isomorphic(H)
- G.is_vertex_transitive()
- G.canonical_label()
- G.minor(graph of minor to find)

تحلیل کلیک ها

- G.is_clique([vertices])
- G.cliques_vertex_clique_number(vertices=[(0, 1), (1, 2)],algorithm="networkx")

networkx can be replaced with cliquer.

- G.cliques_number_of()
- G.cliques_maximum()
- G.cliques_maximal()
- G.cliques_get_max_clique_graph()
- G.cliques_get_clique_bipartite()
- G.cliques_containing_vertex()
- G.clique_number(algorithm="cliquer")
- cliquer can be replaced with networkx.
- G.clique_maximum()
- G.clique_complex()

همبندی در گرافها

- G.is_connected()
- G.connected_component_containing_vertex(vertex)
- G.connected_components_number()
- G.connected_components_subgraphs()
- G.strong_orientation()
- G.strongly_connected_components()
- G.strongly_connected_components_digraph()
- G.strongly_connected_components_subgraphs()
- G.strongly_connected_component_containing_vertex(vertex)
- G.is_strongly_connected()

مسائل NP

- G.vertex_cover(algorithm='Cliquer')

پوشش رأسی

توجه داشته باشید که می توانید از الگوریتم MILP(mixed integer linear program)
نیز استفاده کنید (بایستی بسته GLPK یا CBC را فراخوانی کنید)

- G.hamiltonian_cycle()

یافتن دور همیلتونی

- G.traveling_salesman_problem()

مسأله فروشنده دوره گرد

تغییرات به صورت گرافیکی در گراف

```
g = Graph({0:[1,2,3],1:[0],2:[0,4,4],3:[0,4],4:[2,2,3,6],5:[6],6:[4,8,5],7:[],8:[6]})
g.set_pos({0:[79,139],1:[45.27151527862789,44.88265969424094],2:[183.85254060219972,296.907067626753 9],3:[284.3619563187783 7,145.64958169162242],4:[226,193],5:[256.01483524384605,301.85568288000184],6:[242.25396569173387,239.90403365361175],7:[273,221],8:[293.1063756827343,303.8332652508823]})
graph_editor(g)
```

- اولین دستور گراف را تعریف می کند.
- دومین دستور موقعیت هر رأس را مشخص می نماید.
- سومین دستور ویرایشگر را فراخوانی می کند.

توجه کنید می توانید با یک گراف ساده شروع کنید و گراف مورد نظر خود را مرحله به مرحله بسازید

تغییرات به صورت گرافیکی در گراف

- با دَبل کلیک روی یک فضای خالی می توانید یک رأس جدید اضافه کنید.
- رأسها را می توانید با موشواره حرکت دهید
- با انتخاب یک رأس و سپس رأس دیگر می توانید یال مورد نظر خود را بکشید.
- با انجام مرحله قبل روی یک یال از پیش تعریف شده یال مورد نظر حذف می شود.
- اگر یک رأس تنها پس از حذف یالش حذف می شود، با نگهداشتن کلید shift از اینکار جلوگیری کنید (البته اگر خواستید!)
- با دوبار کلیک کردن روی یک رأس، آن را حذف کرده اید
- با زدن دکمه save گراف مورد نظر شما داخل سلولی که در ابتدا گراف را در آن تعریف کرده بودید قابل مشاهده است.

تهیه جدول

تمام گرافهای با حداکثر ۷ راس در این جدولها قابلا بازخوانی هستند.

```
gdb = GraphDatabase()
graph_db_info(tablename='graph_data')
```

['complement_graph6', 'eulerian', 'graph6', 'lovasz_number', 'num_cycles', 'num_edges', 'num_hamiltonian_cycles', 'num_vertices', 'perfect', 'planar']

تهیه جدولی از گرافها با کمتر مساوی یال و حداقل درجه ۱ مرتب بر اساس درجه

```
Q =GraphQuery(display_cols=['graph6','num_vertices','degree_sequence'],
     num_edges=['<=',5],min_degree=1)
Q.show(with_picture=True)
Q.number_of()
```

تمرینات

- (تحویلی) ۲۰ تمرین از کتاب نظریه گراف

Graph Theory , Bondy & Morty

را با استفاده از Sage حل کنید. (Worksheet را با نام شماره دانشجویی خود ذخیره کنید و توجه کنید هر مسئله در یک سلول باشد. همچنین متن سوال را در ابتدای هر سلول به صورت comment بیاورید) (هر بخش یک تمرین)

پروژه

- بسته ای بنویسید که در sage اعمال به توان رساندن و subdivision از مرتبه دلخواه را انجام دهد.