

Questão 1

Ainda não respondida

Vale 1,00 ponto(s).

🚩 Marcar questão

⚙ Editar questão

Calcule a modularidade para a rede Jazz usando método fastgreedy.

Use:

```
G= nx.read_edgelist("data/jazz.txt", nodetype=int)
G = G.to_undirected()
G.remove_edges_from(nx.selfloop_edges(G))
Gcc = sorted(nx.connected_components(G), key=len, reverse=True)
G = G.subgraph(Gcc[0])
G = nx.convert_node_labels_to_integers(G, first_label=0)
```

- ☐ a. 0.98
- ☐ b. 0.44
- ☐ c. 0.05
- ☐ d. 0.76
- ☐ e. 0.12

Questão 2

Ainda não respondida

Vale 1,00 ponto(s).

🚩 Marcar questão

⚙ Editar questão

Calcule a modularidade para a rede Jazz usando método Louvain.

Use:

```
G= nx.read_edgelist("data/jazz.txt", nodetype=int)
G = G.to_undirected()
G.remove_edges_from(nx.selfloop_edges(G))
Gcc = sorted(nx.connected_components(G), key=len, reverse=True)
G = G.subgraph(Gcc[0])
G = nx.convert_node_labels_to_integers(G, first_label=0)
```

- ☐ a. 0.05
- ☐ b. 0.98
- ☐ c. 0.12
- ☐ d. 0.65
- ☐ e. 0.44

Questão 3

Ainda não respondida

Vale 1,00 ponto(s).

🚩 Marcar questão

⚙ Editar questão

Considere o método de geração de redes LFR_benchmark_graph. Obtenha os valores da modularidade para $\mu = 0.05$, $\mu = 0.1$, $\mu = 0.2$. Use o código a seguir para gerar as redes. Use o algoritmo de Louvain.

`N = 128`

`tau1 = 3`

`tau2 = 1.5`

`k = 16`

`minc = 32`

`maxc = 32`

`G = nx.LFR_benchmark_graph(n = N, tau1 = tau1, tau2 = tau2, mu = mu, min_degree = k, max_degree = k, min_community=minc, max_community = maxc, seed = 10)`

- ☐ a. $Q(\mu=0.05) = 0.64$; $Q(\mu=0.1) = 0.54$; $Q(\mu=0.2) = 0.44$;
- ☐ b. $Q(\mu=0.05) = 0.98$; $Q(\mu=0.1) = 0.71$; $Q(\mu=0.2) = 0.54$;
- ☐ c. $Q(\mu=0.05) = 0.32$; $Q(\mu=0.1) = 0.20$; $Q(\mu=0.2) = 0.04$;
- ☐ d. $Q(\mu=0.05) = 0.54$; $Q(\mu=0.1) = 0.44$; $Q(\mu=0.2) = 0.21$;
- ☐ e. $Q(\mu=0.05) = 0.45$; $Q(\mu=0.1) = 0.24$; $Q(\mu=0.2) = 0.12$;

Questão 4

Ainda não respondida

Vale 1,00 ponto(s).

🚩 Marcar questão

⚙ Editar questão

Considere o método de geração de redes LFR_benchmark_graph. Obtenha os valores da modularidade para $\mu = 0.05$, $\mu = 0.2$, $\mu = 0.4$. Use o código a seguir para gerar as redes. Use o algoritmo de Louvain.

`N = 128`

`tau1 = 3`

`tau2 = 1.5`

`k = 16`

`minc = 32`

`maxc = 32`

`G = nx.LFR_benchmark_graph(n = N, tau1 = tau1, tau2 = tau2, mu = mu, min_degree = k, max_degree = k, min_community=minc, max_community = maxc, seed = 10)`

- ☐ a. $Q(\mu=0.05)=0.76$; $Q(\mu=0.2)=0.33$; $Q(\mu=0.4)=0.23$
- ☐ b. $Q(\mu=0.05)=0.64$; $Q(\mu=0.2)=0.43$; $Q(\mu=0.4)=0.16$
- ☐ c. $Q(\mu=0.05)=0.99$; $Q(\mu=0.2)=0.76$; $Q(\mu=0.4)=0.54$
- ☐ d. $Q(\mu=0.05)=0.88$; $Q(\mu=0.2)=0.62$; $Q(\mu=0.4)=0.44$
- ☐ e. $Q(\mu=0.05)=0.21$; $Q(\mu=0.2)=0.13$; $Q(\mu=0.4)=0.06$