

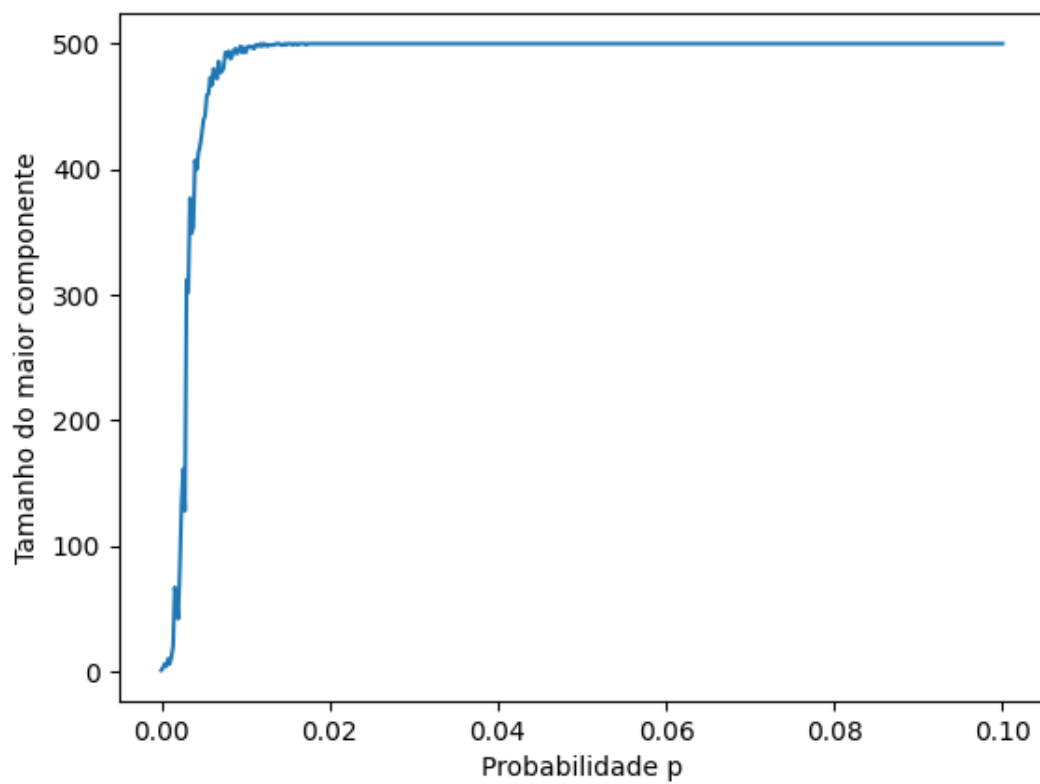
Percolação em grafo aleatório

Jefter Santiago

23 de maio de 2022

Analisar o comportamento do tamanho do componente gigante em uma rede com quantidade de nós igual a N usando o modelo de Erdos-Rényi para gerar grafos aleatórios com probabilidades de p crescentes tal que $p \cdot N = 1$.

```
1  import networkx as nx
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import time
5
6  # numero de nós no grafo
7  n = 500
8  # Lista de probabilidades p de criação de arestas.
9  p = np.linspace(0, 0.1, n)
10
11 # Lista com tamanho dos maiores componentes
12 largest_cc = []
13 start_time = time.time()
14 # Cria grafo de Erdos-Renyi para cada probabilidade p e
15 # armazena o valor do maior componente na lista.
16 for i in range(n):
17     # Lista de grafos aleatorios com probabilidade p.
18     er = nx.gnp_random_graph(n,p[i])
19     # Tamanho do maior componente para cada probabilidade.
20     largest_cc.append(len(max(nx.connected_components(er), key=len)))
21
22 total_time = time.time() - start_time
23 print("For n = ", n)
24 print("-- %s seconds --" % total_time )
25
26 plt.plot(p, largest_cc)
27 plt.xlabel("Probabilidade p")
28 plt.ylabel("Tamanho do maior componente")
29 plt.show()
```



Tempo de execução: $f(n) = \sum_{i=0}^n \sum_{j=0}^n = \sum_{i,j=0}^n \Rightarrow f(n) = O(n^2)$