

15/04/2025

Analyse de réseaux

« Bitcoin Transactions » et
« Madoff Fraud »



Ozren PAIN
L3 ECONOMIE ET GESTION

Analyse du réseau Bitcoin Transactions

Ce réseau représente les transactions du bitcoin couvrant la période de janvier 2009 à avril 2013. Les nœuds sont des « adresses » à clé publique à un moment donné dans la période donnée, et chaque bord est marqué avec l'heure (à une résolution de quelques secondes) de la transaction. Cela permet de retracer avec précision la chronologie des échanges, ce qui est fondamental pour comprendre les dynamiques temporelles et l'évolution de l'activité économique au sein du réseau. Le jeu de données est fourni sous la forme de deux fichiers CSV : l'un contenant les liens (avec l'adresse source, l'adresse cible, le nombre de transactions, et les dates extrêmes associées), l'autre listant les nœuds, chacun associé à la date de sa première transaction. Ce type de réseau est particulièrement intéressant dans une perspective d'analyse économique et systémique car il représente les flux réels de valeur dans un système décentralisé sans entité centrale de régulation, mettant ainsi en lumière les dynamiques propres aux échanges monétaires dans les cryptoéconomies.

Le réseau complet est très vaste, avec environ 1 000 000 d'adresses (nœuds) et plus de 1 000 000 d'arêtes (transactions). Face à cette ampleur, et en raison des limites computationnelles des environnements standards comme Google Colab, il a fallu opter pour une approche mixte. D'un côté, certaines mesures globales ont été extraites de la composante géante du réseau, c'est-à-dire le sous-graphe contenant la plus grande partie des nœuds connectés ensemble (environ 519 000 nœuds sur plus d'un million, soit près de 42 % du réseau total). D'un autre côté, pour les métriques particulièrement gourmandes en ressources telles que la centralité de vecteur propre, le coefficient de clustering, la détection de communautés ou l'analyse des triangles, il a fallu extraire un échantillon aléatoire représentatif de 2000 nœuds appartenant à cette composante géante. Cela permet de préserver la validité statistique de l'analyse tout en garantissant des temps de calcul raisonnables.

L'analyse structurelle du réseau révèle un graphe extrêmement peu dense. Le calcul de la densité, c'est-à-dire le rapport entre le nombre d'arêtes existantes et le nombre maximum théorique d'arêtes possibles, donne une valeur proche de zéro. Ce résultat est cohérent avec la nature d'un réseau transactionnel dans un système où la plupart des adresses ne sont utilisées qu'une seule fois, ou très ponctuellement, ce qui est une caractéristique bien documentée dans les recherches sur Bitcoin. Cette faible densité reflète une économie de pair-à-pair où la confidentialité prime, entraînant des structures en étoile autour de quelques acteurs et une multitude d'interactions uniques. La majorité des adresses jouent un rôle marginal dans le réseau, ce qui se reflète aussi dans les distributions de degrés. Ces dernières montrent une forte asymétrie :

beaucoup d'adresses n'ont qu'un seul lien entrant ou sortant, tandis qu'une minorité concentre de nombreux liens, ce qui donne au réseau une structure hétérogène.

La centralité de degré permet d'identifier ces nœuds particuliers qui reçoivent ou émettent un grand nombre de transactions. Sur l'échantillon de 2000 nœuds, plusieurs adresses apparaissent comme des hubs. Toutefois, lorsque l'on s'intéresse à la centralité d'intermédiarité, qui mesure l'importance d'un nœud dans le cheminement des flux entre d'autres nœuds, on observe que la plupart des adresses n'ont pas de rôle de médiation. Le score d'intermédiarité est souvent nul, ce qui confirme que le réseau est peu hiérarchisé, avec une structure très plate. Ce constat est renforcé par la centralité de proximité, qui reste relativement homogène dans l'échantillon, et la centralité de vecteur propre, qui ne révèle pas de figures dominantes. Cela indique une forme de décentralisation effective, où l'activité économique est distribuée sans concentration excessive de pouvoir transactionnel.

D'un point de vue topologique, le réseau Bitcoin présente certaines caractéristiques des réseaux de type "petit monde". La longueur moyenne des plus courts chemins dans la composante géante est d'environ 3,9, et le diamètre atteint 9. Ces valeurs sont faibles au regard de la taille du réseau, indiquant que pour peu qu'un chemin existe, les transactions peuvent circuler rapidement entre les nœuds. Cette faible distance moyenne favorise une certaine efficacité transactionnelle, malgré la fragmentation apparente. Toutefois, cette efficacité apparente ne doit pas masquer le fait que le réseau est constitué d'un grand nombre de petites composantes non connectées à la composante géante, ou de nœuds très faiblement liés. De ce point de vue, la structure globale reste très morcelée.

Le réseau Bitcoin semble également extrêmement fragile. Par exemple, en mesurant les 10 % des nœuds ayant les plus hauts degrés et en mesurant la taille relative de la composante géante restante. Celle-ci chute à environ 7 % du réseau initial, ce qui indique une dépendance structurelle à une minorité de nœuds très connectés. Cette propriété est typique des réseaux à distribution de degré suivant une loi de puissance, ce qui semble être le cas ici si l'on en juge par la forme des histogrammes en échelle log-log. Cela implique que, si un acteur malveillant parvenait à isoler ou neutraliser les quelques adresses les plus connectées, le réseau dans son ensemble pourrait se retrouver fortement fragmenté, ce qui pose la question de la résilience de la blockchain dans son usage transactionnel. Cette fragilité structurelle suggère que l'illusion de décentralisation peut dissimuler une forte dépendance à quelques acteurs-clés du système.

L'analyse des communautés permet d'aller plus loin dans la compréhension de la modularité du réseau. En appliquant sur l'échantillon de 2000 nœuds une méthode de détection basée sur la modularité (greedy modularity communities), on trouve près de 2000 communautés distinctes. Cela signifie que, sur cet échantillon, quasiment chaque adresse est isolée dans sa propre sous-structure, ou forme un tout petit groupe avec quelques autres adresses. Cela reflète la nature éphémère et anonyme des transactions Bitcoin, où les utilisateurs génèrent souvent de nouvelles adresses pour chaque transaction afin d'accroître leur anonymat. Le fait que le noyau le plus dense (k-core maximal) du réseau soit égal à 1 confirme cette faible densité structurelle : il n'existe pas de sous-réseau dans lequel chaque nœud serait connecté à au moins deux autres, ce qui tranche avec les réseaux sociaux classiques, où les communautés forment souvent des noyaux denses. Cette modularité extrême illustre un modèle d'interaction économique sans engagement durable, propre aux logiques d'échange automatisé et pseudonyme.

Les autres métriques confirment ce portrait : le degré moyen des voisins (k-nearest neighbors) est très faible, souvent nul, ce qui suggère que les adresses ne se connectent pas à des clusters mais à des cibles isolées. Le nombre de triangles détectés est lui aussi très faible, ce qui témoigne d'un très faible taux de transitivité et d'une absence de redondance dans les connexions. Autrement dit, les adresses échangent rarement avec plusieurs partenaires qui échangent aussi entre eux, ce qui est contraire à ce qu'on observe dans des réseaux de type social ou professionnel. Ce manque de redondance traduit une structure transactionnelle cloisonnée, adaptée à une logique de micro-échange ponctuel plutôt qu'à des dynamiques collectives.

Enfin, bien qu'il n'ait pas été possible de visualiser le réseau complet en raison de sa taille, les propriétés structurelles observées sur la composante géante et l'échantillon aléatoire permettent d'en déduire des caractéristiques visuelles probables. Par exemple, la structure en étoile autour de quelques nœuds très connectés, ainsi que la multitude de nœuds faiblement liés ou isolés, laissent supposer un graphe globalement éclaté, sans clusters denses ni structures communautaires marquées. Cette configuration illustre parfaitement la pratique courante du "one-time use address" dans l'écosystème Bitcoin. Même sans visualisation directe, les indicateurs structurels pointent vers un réseau atomisé, où les interactions sont dispersées, peu redondantes, et rarement récurrentes.

Cette analyse du réseau Bitcoin Transactions entre 2009 et 2013 révèle donc un système transactionnel vaste mais profondément éclaté, faiblement connecté et faiblement redondant. La combinaison d'une faible densité, d'une modularité extrême, d'une grande vulnérabilité structurelle, et d'une hiérarchie aplatie rend ce réseau

unique en son genre. Il ne s'agit ni d'un réseau social, ni d'un réseau économique classique, mais bien d'un artefact numérique issu d'une logique de confidentialité, d'instantanéité, et d'automatisation des échanges. Ce réseau ne reflète pas une communauté d'utilisateurs connectés entre eux, mais plutôt une succession de micro-interactions sans mémoire collective, sans redondance, et sans structure stable à long terme. Il incarne un modèle économique d'échange fluide mais impersonnel, propre à une vision radicale de la décentralisation.

Analyse du réseau Madoff

Le second réseau analysé dans le cadre de ce devoir porte sur les flux financiers impliqués dans la fraude de Bernard Madoff, l'une des plus grandes escroqueries financières de l'histoire contemporaine. Madoff, ancien président du NASDAQ et gestionnaire de fonds d'investissement, a été reconnu coupable d'avoir orchestré un schéma de type Ponzi d'une ampleur sans précédent, entraînant la perte de dizaines de milliards de dollars pour des particuliers, des banques et des institutions à travers le monde. Le réseau étudié ici a été reconstitué à partir de sources publiques, notamment des articles de presse, des documents judiciaires et des enquêtes médiatiques, et il représente les flux d'argent entre les entités financières ayant investi, directement ou indirectement, dans le fonds de Madoff.

Le réseau est constitué de 61 nœuds, représentant des institutions financières, banques privées, sociétés de gestion et fonds d'investissement. Il s'agit d'un graphe dirigé, où chaque arête correspond à un flux monétaire d'un acteur vers un autre. Le réseau est dense en information malgré sa taille modeste, car il reflète un enchevêtrement complexe de relations financières, de délégations de gestion, et de montages financiers visant à canaliser l'argent des investisseurs vers l'entité centrale du schéma frauduleux : Bernard_Madoff_Investment. Contrairement au réseau Bitcoin, où les liens peuvent être temporaires et massivement distribués, ici les relations sont plus stables, structurelles et intentionnelles, ce qui reflète une dynamique de réseau institutionnel très organisée.

Les propriétés globales du réseau confirment cette structure hiérarchisée. Le graphe est faiblement dense (densité d'environ 0,033), mais néanmoins connecté : tous les nœuds sont liés dans une seule composante. Cela signifie que, malgré une faible proportion de liens par rapport au maximum théorique, l'ensemble du système reste unifié autour d'un centre névralgique. Le degré moyen est de 2, ce qui est relativement faible, mais certaines institutions présentent des degrés bien plus élevés, à commencer par Bernard_Madoff_Investment, qui détient à lui seul un degré centralisé

de 0.45, soit de très loin le plus élevé du réseau. Cela confirme immédiatement sa position de point de convergence de tous les flux financiers.

L'analyse des centralités va dans le même sens. En plus d'être le nœud le plus connecté, Bernard_Madoff_Investment se situe au cœur du réseau en termes de centralité d'intermédiation : les liens qui mènent vers lui, notamment ceux émanant de Cohmad_securities, Fairfield_Greenwich, Tremont_Group_Holdings ou Gabriel_Capital, possèdent les scores d'intermédiation les plus élevés du réseau. Ces connexions ne sont pas anodines : elles représentent en réalité des "feeder funds", c'est-à-dire des fonds spécialisés qui collectaient de l'argent auprès de multiples clients ou d'autres institutions avant de le transférer directement à Madoff. En ce sens, ces fonds jouaient un rôle de courroies de transmission, dissimulant parfois à leurs propres clients la véritable destination des capitaux. Le rôle structurant de ces nœuds intermédiaires dans le fonctionnement du réseau est flagrant, et leur suppression aurait un impact majeur sur la connectivité du graphe.

Cela est précisément ce que montre l'analyse de la robustesse structurelle du réseau. En simulant la suppression des 10 % de nœuds ayant les degrés les plus élevés, on observe que la taille de la composante principale chute brutalement à 6,5 % du réseau initial. Le réseau est donc extrêmement vulnérable à des attaques ciblées, confirmant que sa structure repose sur une minorité critique d'intermédiaires. Cette propriété est typique des réseaux frauduleux centralisés : leur apparente diversité masque en réalité une architecture en étoile autour d'un centre unique, ici Bernard Madoff, et d'une poignée de nœuds pivot.

L'analyse des plus courts chemins et du diamètre ($APL \approx 3,43$, diamètre = 7) montre une certaine efficacité dans la circulation des flux, renforçant l'idée que le réseau était conçu pour canaliser rapidement les fonds vers Madoff, quelle que soit leur provenance initiale. Toutefois, contrairement à un réseau social ou transactionnel "naturel", ici cette efficacité n'est pas le fruit d'interactions organiques, mais d'une construction volontairement orientée vers un seul but : masquer la fraude derrière un rideau de complexité apparente.

Les mesures de clustering viennent confirmer cette absence d'interactions horizontales. Le coefficient de clustering moyen est nul, et aucun triangle n'a été détecté dans le réseau. Cela signifie qu'il n'y a pas de groupes d'acteurs interconnectés entre eux, pas de redondance relationnelle. Chaque lien va d'un investisseur à un intermédiaire, ou d'un intermédiaire à Madoff, mais très rarement d'un investisseur à un autre, ou d'un fonds à plusieurs autres. Cette absence de

fermeture des triplets reflète un manque total de contrôle croisé entre les acteurs : personne ne vérifie ce que fait l'autre, et chacun opère dans une relative opacité, ce qui a sans doute contribué à la longévité de la fraude.

La détection de communautés a permis d'identifier 10 modules dans le réseau. Chaque communauté correspond globalement à un canal de collecte de fonds distinct, souvent centré sur un ou deux acteurs pivot. Cela souligne la stratégie de segmentation de Madoff : plutôt que d'accumuler tous les flux dans un seul chemin, il a multiplié les circuits parallèles, ce qui a permis d'élargir sa base de collecte tout en dispersant les responsabilités. De manière significative, ces communautés ne présentent pas de forte densité interne : comme l'indique le k-core maximal (égal à 1), il n'existe aucun sous-ensemble de nœuds fortement interconnectés entre eux. Cela révèle une absence totale de structure communautaire au sens fort : les fonds sont liés verticalement à Madoff, mais pas horizontalement entre eux.

Enfin, la structure du réseau est renforcée par une forte désassortativité de degré (≈ -0.37). Cela signifie que les nœuds de degré élevé (notamment Madoff et ses proches collaborateurs) sont principalement connectés à des nœuds de faible degré, c'est-à-dire des entités isolées qui n'ont pas d'autres connexions. Cette architecture accentue la centralisation du système et isole chaque flux, réduisant les risques de contrôle mutuel ou d'alerte en cas de dysfonctionnement. Ce type de structuration est une signature classique des schémas Ponzi, dans lesquels un petit nombre d'entités récoltent les capitaux d'un grand nombre d'investisseurs déconnectés, en leur promettant des rendements fictifs.

L'analyse de ce réseau met ainsi en évidence le lien étroit entre la topologie d'un graphe financier et le fonctionnement d'un schéma de fraude. Le réseau Madoff, derrière sa simplicité apparente (seulement 61 nœuds), révèle une ingénierie organisationnelle sophistiquée, fondée sur la centralisation des flux, la fragmentation des collecteurs, et l'absence de transparence. Chaque métrique réseau vient confirmer ce constat : faible densité, absence de triangles, centralité extrême de quelques nœuds, modularité sans densité, fragilité à la suppression ciblée. L'ensemble dessine une architecture quasi parfaite pour un Ponzi scheme : un système qui repose sur la confiance aveugle, l'éclatement des responsabilités, et la difficulté d'avoir une vision globale des flux.

Comparaison des réseaux Bitcoin et Madoff

L'analyse comparative des réseaux Bitcoin (2009-2013) et Madoff révèle deux architectures radicalement opposées, tant par leur structure que par leur finalité. Le réseau Bitcoin, issu d'une technologie décentralisée, se caractérise par une topologie éclatée faiblement connectée et peu redondante. À l'inverse, le réseau Madoff, construit autour d'un schéma frauduleux présente une structure centralisée hiérarchisée et hautement vulnérable à des attaques ciblées.

Le réseau Bitcoin est marqué par une faible densité et une absence de redondance, reflétant la nature à la fois éphémère et anonyme des transactions. Les adresses sont souvent utilisées une seule fois, et les interactions entre elles sont limitées. Cette structure favorise la confidentialité ainsi que sa résilience, mais complique l'analyse des flux financiers. En revanche, le réseau Madoff est dense et centralisé avec des flux financiers convergeant vers une entité unique. Cette centralisation facilite la gestion des fonds mais rend le système extrêmement vulnérable à la défaillance de nœuds clés.

Sur le plan des centralités, le réseau Bitcoin ne présente pas de nœuds dominants, ce qui indique une distribution relativement équitable des interactions. À l'opposé, le réseau Madoff est dominé par quelques nœuds centraux, notamment Bernard Madoff Investment Securities qui concentrent l'essentiel des flux financiers. Cette concentration est typique des schémas de Ponzi, où les fonds des nouveaux investisseurs sont utilisés pour payer les rendements des anciens sans véritable activité économique sous-jacente.

La détection de communautés dans le réseau Bitcoin révèle une multitude de petits groupes faiblement connectés, sans structure hiérarchique claire. Cela contraste avec le réseau Madoff, où les communautés sont structurées autour de fonds d'investissement spécifiques, jouant le rôle d'intermédiaires entre les investisseurs et Madoff. Cette organisation facilite la dissimulation de la fraude et complique la traçabilité des fonds.

Enfin, la résilience des deux réseaux diffère considérablement. Le réseau Bitcoin est fragmenté mais est robuste face à la suppression de nœuds, grâce à sa nature décentralisée. En revanche, le réseau Madoff est extrêmement fragile : la suppression de quelques nœuds centraux suffit à désintégrer l'ensemble du système. Cette fragilité est inhérente aux schémas de Ponzi, qui reposent sur la confiance et la continuité des flux entrants.

L'analyse comparative de ces deux réseaux met donc en lumière les différences fondamentales entre un système décentralisé conçu pour la résilience et la confidentialité, et un système centralisé structuré autour d'une fraude, ce qui illustre donc l'importance de la topologie des réseaux dans la compréhension des dynamiques économiques et des risques associés.

Annexes

1) Bitcoin transactions

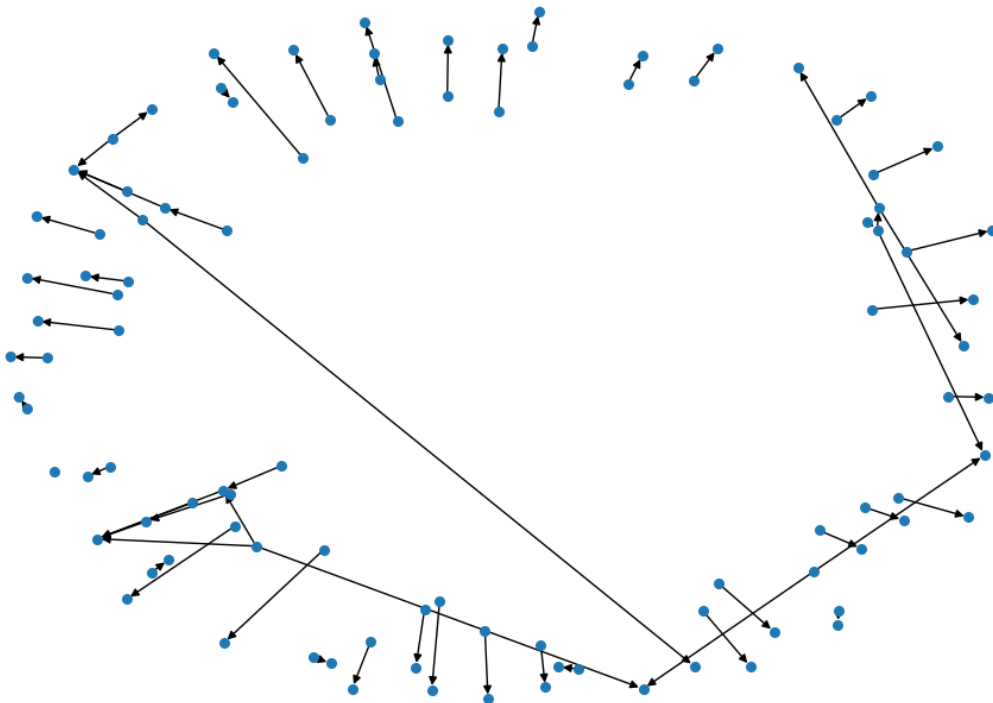
```
print("Nombre de noeuds :", G.number_of_nodes())  
print("Nombre d'arêtes :", G.number_of_edges())
```

```
Nombre de noeuds : 1229276  
Nombre d'arêtes : 1063060
```

```
attributes = pd.Series(vertices_df.first_transaction_date.values, index=vertices_df.vid).to_dict()  
nx.set_node_attributes(G, attributes, 'first_transaction_date')
```

```
sample_nodes = list(G.nodes())[:100]  
subG = G.subgraph(sample_nodes)  
pos = nx.spring_layout(subG, seed=42)  
plt.figure(figsize=(10,7))  
nx.draw(subG, pos, with_labels=False, node_size=30, arrowsize=10)  
plt.title("Sous-graphe (100 premiers nœuds) Bitcoin Transactions")  
plt.show()
```

Sous-graphe (100 premiers nœuds) Bitcoin Transactions



```
print("Densité :", nx.density(G))
print("Le graphe est-il fortement connecté ?", nx.is_strongly_connected(G))
print("Le graphe est-il faiblement connecté ?", nx.is_weakly_connected(G))
```

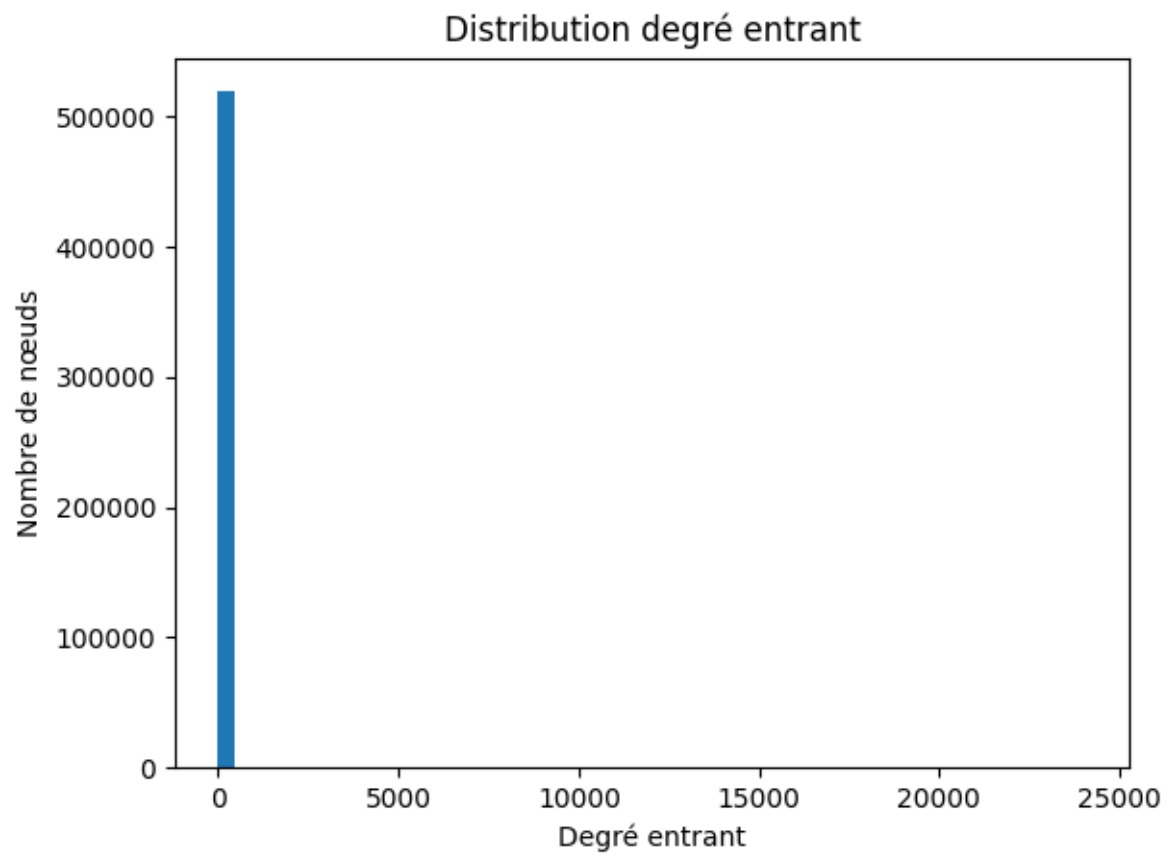
```
Densité : 7.034922624704506e-07
Le graphe est-il fortement connecté ? False
Le graphe est-il faiblement connecté ? False
```

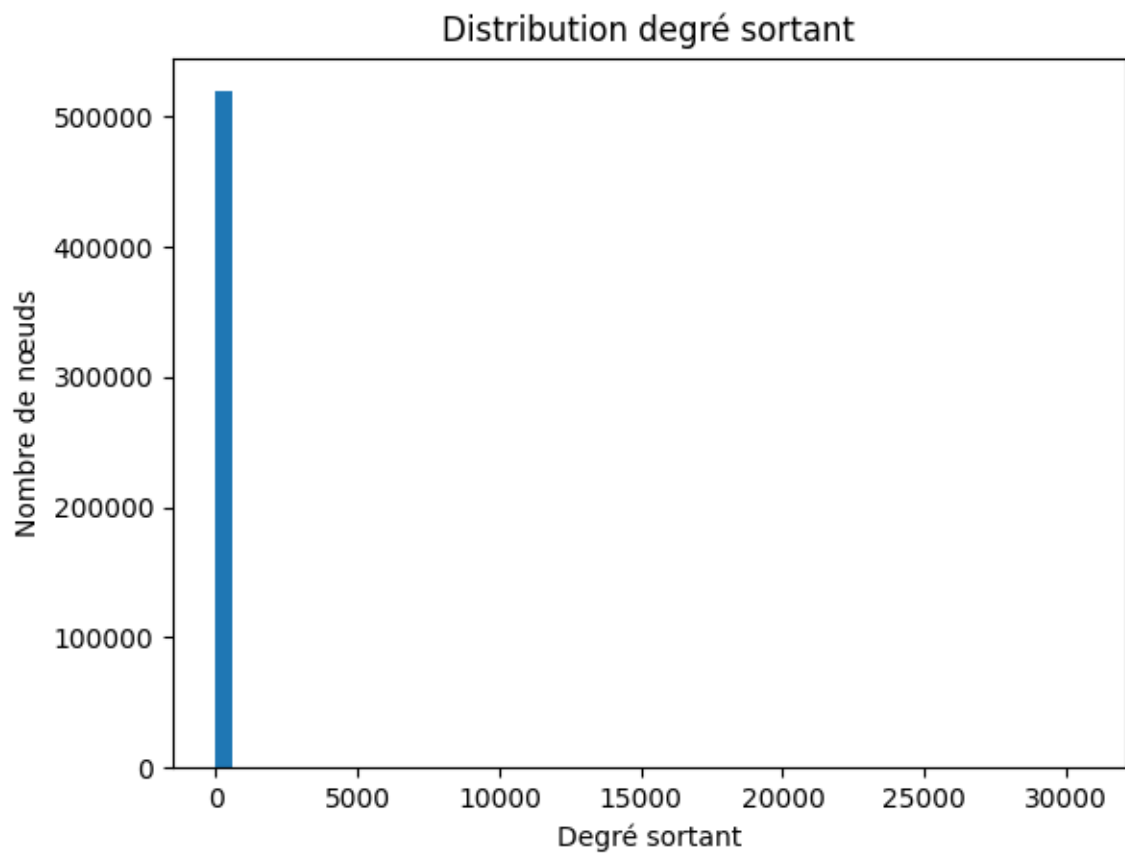
```
giant_component = max(nx.weakly_connected_components(G), key=len)
G_giant = G.subgraph(giant_component).copy()
```

```
print("Taille de la composante géante :", len(G_giant.nodes))
```

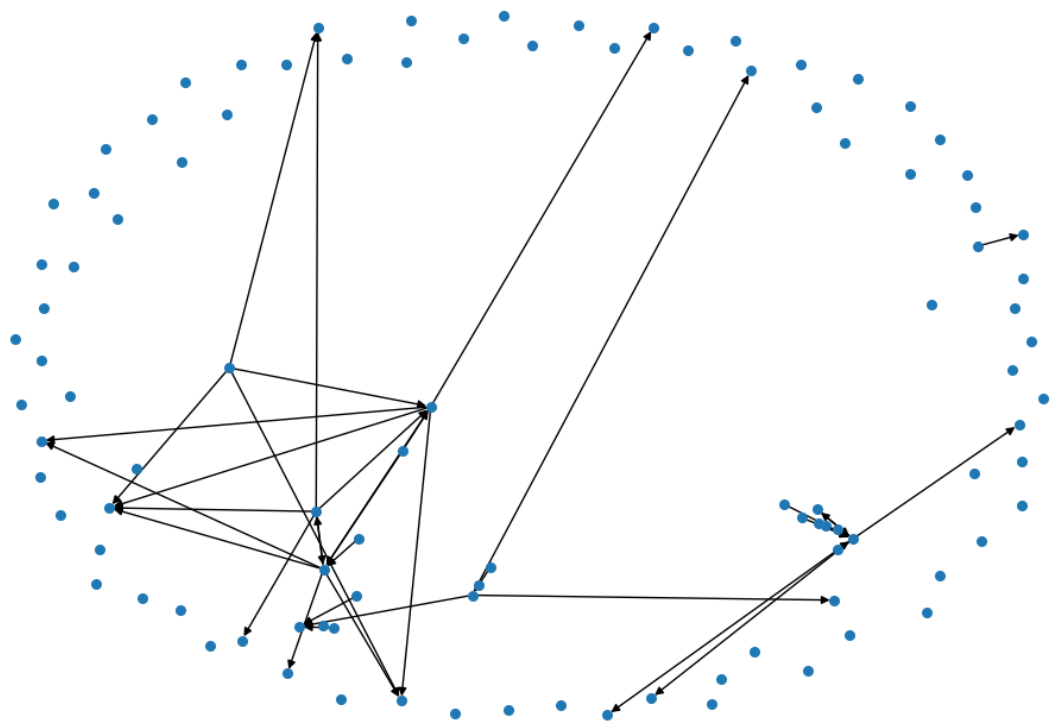
```
Taille de la composante géante : 518969
```

```
sample_nodes = list(G_giant.nodes())[:100]
subG_giant = G_giant.subgraph(sample_nodes)
pos = nx.spring_layout(subG_giant, seed=42)
plt.figure(figsize=(10,7))
nx.draw(subG_giant, pos, with_labels=False, node_size=30, arrowsize=10)
plt.title("Sous-graphe (100 premiers nœuds) composante géante Bitcoin")
plt.show()
```





Sous-graphe (100 premiers nœuds) composante géante Bitcoin



```

in_strength = dict(G_giant.in_degree(weight='weight'))
out_strength = dict(G_giant.out_degree(weight='weight'))

import random

sample_nodes = random.sample(list(G_giant.nodes()), 2000)

G_sample = G_giant.subgraph(sample_nodes).copy()

degree_centrality_sample = nx.degree_centrality(G_sample)
closeness_centrality_sample = nx.closeness_centrality(G_sample)
betweenness_centrality_sample = nx.betweenness_centrality(G_sample)
eigenvector_centrality_sample = nx.eigenvector_centrality(G_sample, max_iter=1000)

print("Top 5 Centralité de degré:", sorted(degree_centrality_sample.items(), key=lambda x: x[1], reverse=True)[:5])
print("Top 5 Centralité de Eigenvectors:", sorted(eigenvector_centrality_sample.items(), key=lambda x: x[1], reverse=True)[:5])

```

```

weights = [d['weight'] for u,v,d in G_giant.edges(data=True)]
plt.hist(weights, bins=50, log=True)
plt.title("Distribution des poids des transactions")
plt.xlabel("Nombre de transactions (poids)")
plt.ylabel("Nombre de liens")
plt.show()

```

```

[ ] top_degree = sorted(degree_centrality_sample.items(), key=lambda x: x[1], reverse=True)[:5]
    print("Top 5 (centralité de degré) :", top_degree)

↕ Top 5 (centralité de degré) : [(np.int64(25), 0.09912364538854034), (np.int64(29), 0.046742766413343406), (np.int64(11), 0.046349678592899755), (np.int64(74), 0.0350021581...)]

[ ] top_degree = sorted(degree_centrality_sample.items(), key=lambda x: x[1], reverse=True)[:5]
    print("Top 5 (centralité de degré) :", top_degree)

↕ Top 5 (centralité de degré) : [(np.int64(25), 0.09912364538854034), (np.int64(29), 0.046742766413343406), (np.int64(11), 0.046349678592899755), (np.int64(74), 0.0350021581...)]

[ ] top_betweenness = sorted(betweenness_centrality_sample.items(), key=lambda x: x[1], reverse=True)[:5]
    print("Top 5 (centralité d'intermédierité) :", top_betweenness)

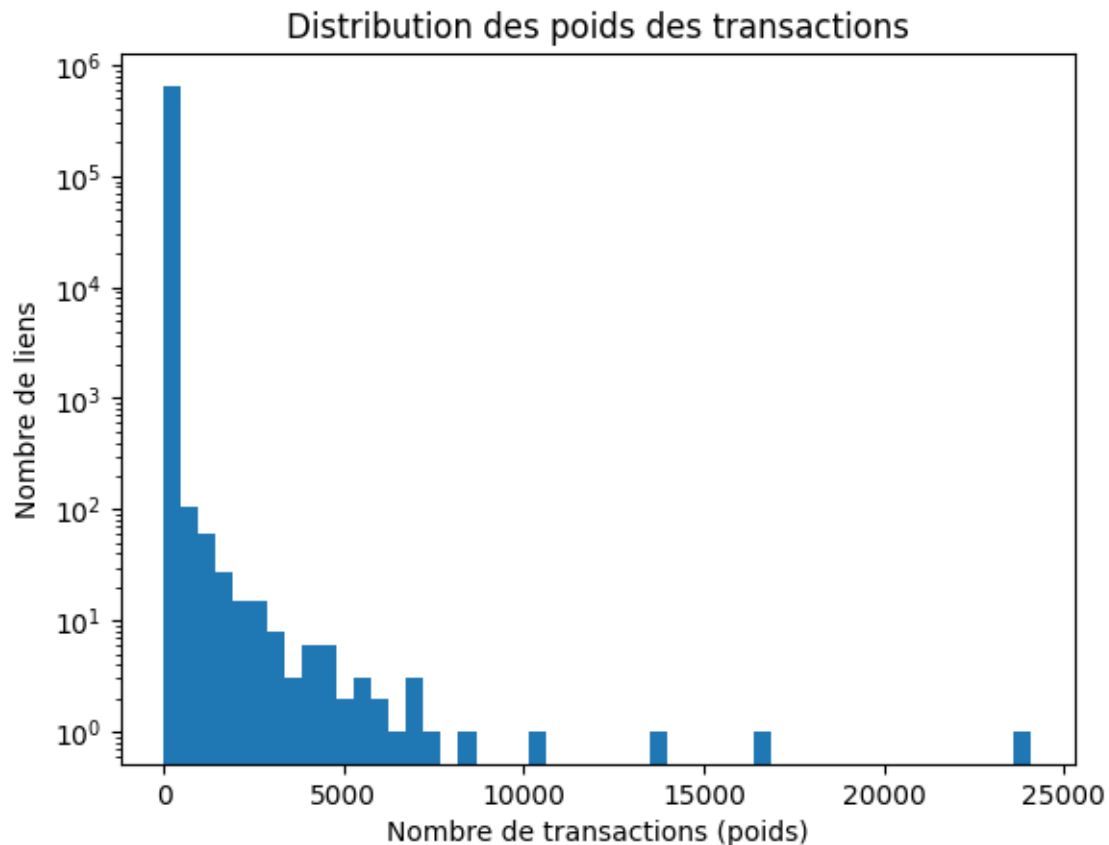
↕ Top 5 (centralité d'intermédierité) : [(np.int64(2179080), 0.0), (np.int64(4833292), 0.0), (np.int64(4472853), 0.0), (np.int64(286759), 0.0), (np.int64(1548329), 0.0)]

```

```

weights = [d['weight'] for u,v,d in G_giant.edges(data=True)]
plt.hist(weights, bins=50, log=True)
plt.title("Distribution des poids des transactions")
plt.xlabel("Nombre de transactions (poids)")
plt.ylabel("Nombre de liens")
plt.show()

```



```
[ ] assortativity = nx.degree_assortativity_coefficient(G_giant)
    print("Assortativité (degré) :", assortativity)
```

➡ Assortativité (degré) : -0.10425006782090754

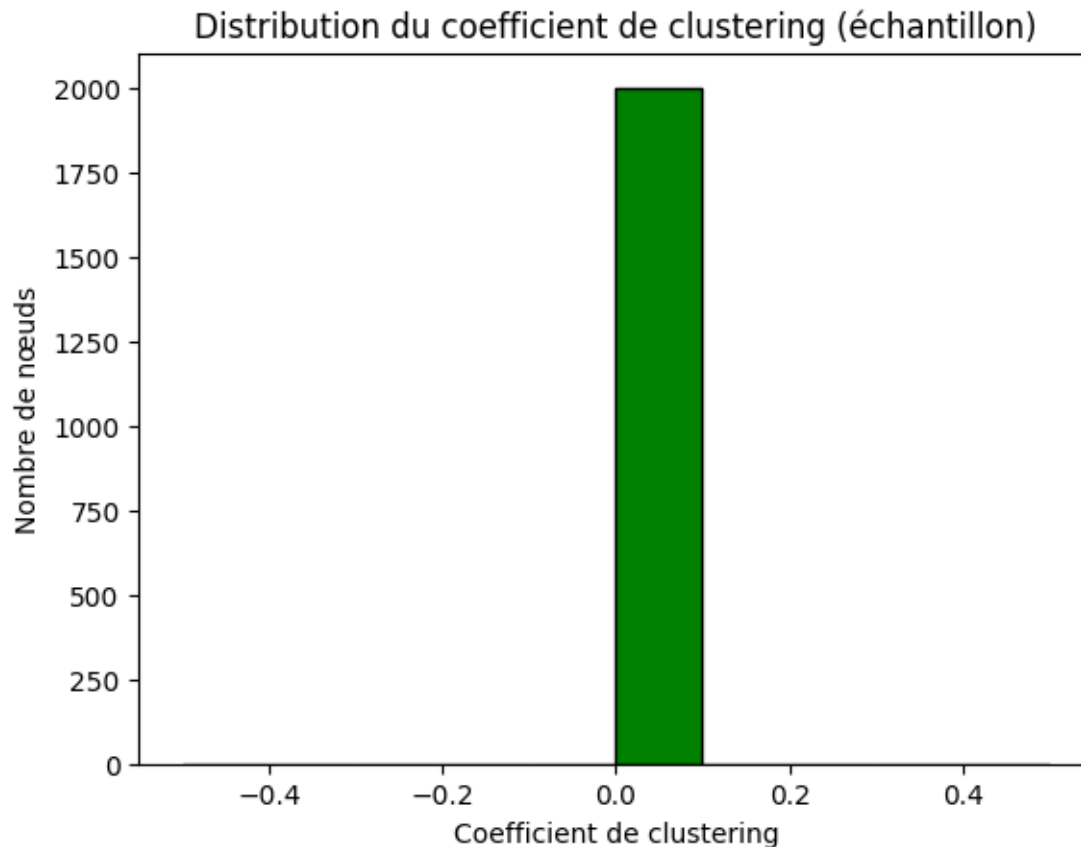
```
sample_nodes = random.sample(list(G_giant.nodes()), 2000)
G_sample = G_giant.subgraph(sample_nodes).copy()

clustering_sample = nx.clustering(G_sample)

avg_clustering_sample = nx.average_clustering(G_sample)

print("Coefficient de clustering moyen (échantillon) :", avg_clustering_sample)

plt.hist(list(clustering_sample.values()), bins=10, color='green', edgecolor='black')
plt.title('Distribution du coefficient de clustering (échantillon)')
plt.xlabel('Coefficient de clustering')
plt.ylabel('Nombre de nœuds')
plt.show()
```



```
largest_scc = max(nx.strongly_connected_components(G_giant), key=len)
G_scc = G_giant.subgraph(largest_scc).copy()

print("Longueur moyenne des chemins (APL) :", nx.average_shortest_path_length(G_scc))
print("Diamètre de la composante géante :", nx.diameter(G_scc.to_undirected()))
```

```
Longueur moyenne des chemins (APL) : 3.9096653729676856
Diamètre de la composante géante : 9
```

```
from copy import deepcopy

def robustness_test(G, removal_fraction=0.1):
    G_copy = deepcopy(G)
    num_remove = int(removal_fraction * G_copy.number_of_nodes())
    nodes_sorted = sorted(G_copy.degree, key=lambda x: x[1], reverse=True)
    to_remove = [n for n, d in nodes_sorted[:num_remove]]
    G_copy.remove_nodes_from(to_remove)
    largest_cc = max(nx.weakly_connected_components(G_copy), key=len)
    return len(largest_cc) / G.number_of_nodes()

print("Taille relative après suppression de 10% des hubs :", robustness_test(G_giant, 0.1))
```

```
Taille relative après suppression de 10% des hubs : 1.7342076308989554e-05
```



```

sample_nodes = random.sample(list(G_giant.nodes()), 2000)
G_sample = G_giant.subgraph(sample_nodes).copy()

G_sample_undirected = G_sample.to_undirected()

communities_sample = nxcom.greedy_modularity_communities(G_sample_undirected)

print("Nombre de communautés détectées (échantillon) :", len(communities_sample))

```

Nombre de communautés détectées (échantillon) : 1995

```

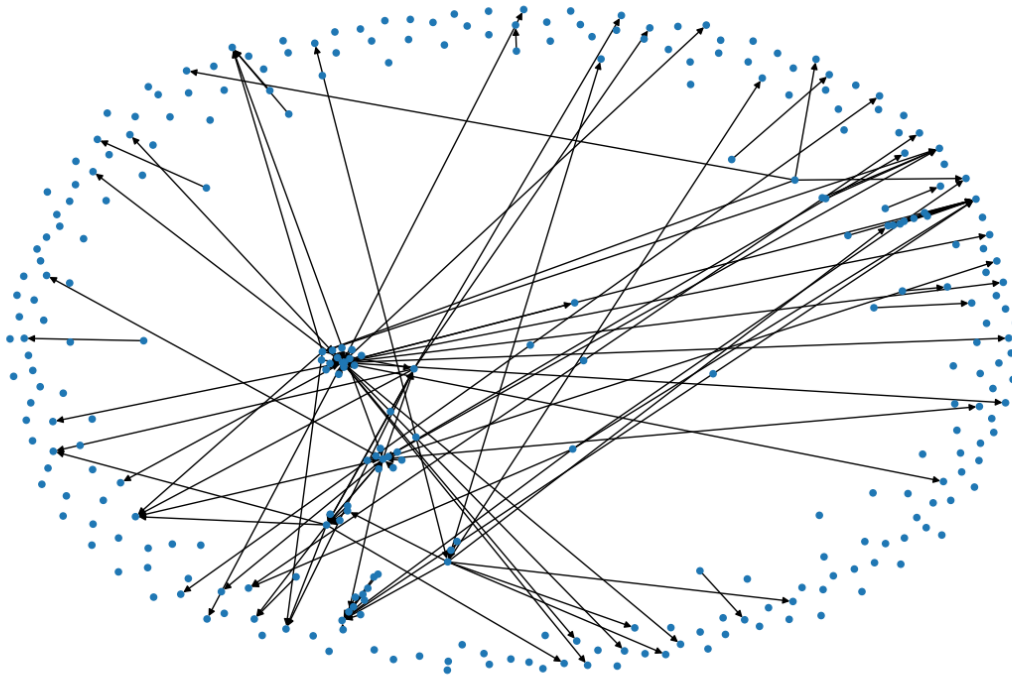
color_map = {}
for c_idx, community in enumerate(communities_sample):
    for name in community:
        color_map[name] = c_idx

colors = [color_map.get(n, 0) for n in G_giant.nodes()]
sample_nodes = list(G_giant.nodes())[:300]
subG_giant = G_giant.subgraph(sample_nodes)
pos = nx.spring_layout(subG_giant, seed=42)

plt.figure(figsize=(12,8))
nx.draw(subG_giant, pos, node_color=[color_map.get(n, 0) for n in sample_nodes],
        with_labels=False, node_size=20, cmap=plt.cm.tab20)
plt.title('Communautés Bitcoin (échantillon)')
plt.show()

```

Communautés Bitcoin (échantillon)



```
sample_nodes = random.sample(list(G_giant.nodes()), 2000)
G_sample = G_giant.subgraph(sample_nodes).copy()
G_sample_undirected = G_sample.to_undirected()

core_numbers_sample = nx.core_number(G_sample_undirected)
print("Noyau le plus dense (k-core max, échantillon) :", max(core_numbers_sample.values()))
```

Noyau le plus dense (k-core max, échantillon) : 1

```
A_sample = nx.adjacency_matrix(G_sample)
print("Matrice d'adjacence du sous-graphe (échantillon) :")
print(A_sample.todense())
```

Matrice d'adjacence du sous-graphe (échantillon) :

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

```

!pip install networkx
import networkx as nx
import numpy as np

knn_sample = nx.average_neighbor_degree(G_sample_undirected)

print("Degré moyen des voisins (échantillon) :", knn_sample)

al/lib/python3.11/dist-packages (3.4.2)
513824): 0.0, np.int64(2187268): 0.0, np.int64(450565): 0.0, np
_

triangles_sample = sum(nx.triangles(G_sample_undirected).values()) // 3
print("Nombre de triangles (échantillon) :", triangles_sample)

Nombre de triangles (échantillon) : 0

```

2) Madoff Fraud

```

import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

madoff_df = pd.read_csv('MADOFF.csv', index_col=0)

# Replace NaN values with 0
madoff_df = madoff_df.fillna(0)

G = nx.from_pandas_adjacency(madoff_df)

print("Nombre de noeuds :", G.number_of_nodes())
print("Nombre d'arêtes :", G.number_of_edges())

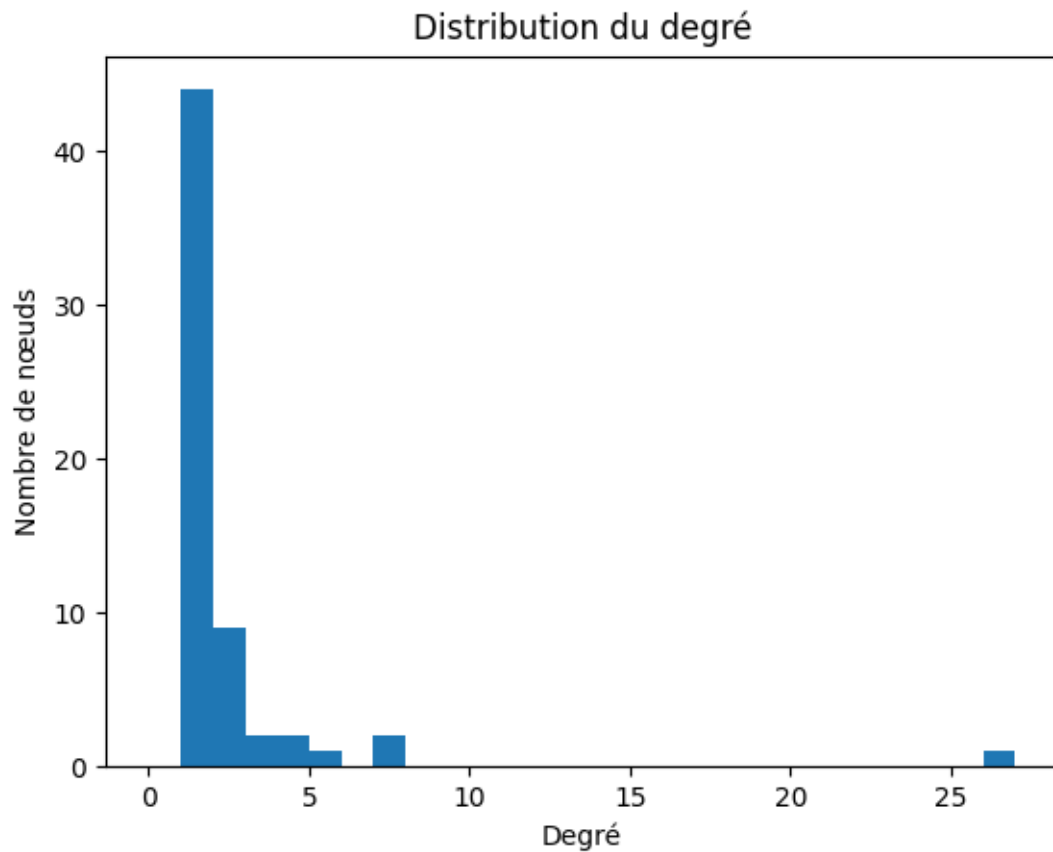
Nombre de noeuds : 61
Nombre d'arêtes : 61

print("Densité :", nx.density(G))

Densité : 0.03333333333333333

```

```
plt.hist(degree_sequence, bins=range(max(degree_sequence)+1))
plt.title("Distribution du degré")
plt.xlabel("Degré")
plt.ylabel("Nombre de nœuds")
plt.show()
```



```
avg_degree = sum(degree_sequence) / G.number_of_nodes()
print("Degré moyen :", avg_degree)
```

Degré moyen : 2.0

```
assortativity = nx.degree_assortativity_coefficient(G)
print("Assortativité par degré :", assortativity)
```

Assortativité par degré : -0.3772451095955949

```
deg centrality = nx.degree centrality(G)

closeness centrality = nx.closeness centrality(G)

betweenness centrality = nx.betweenness centrality(G)

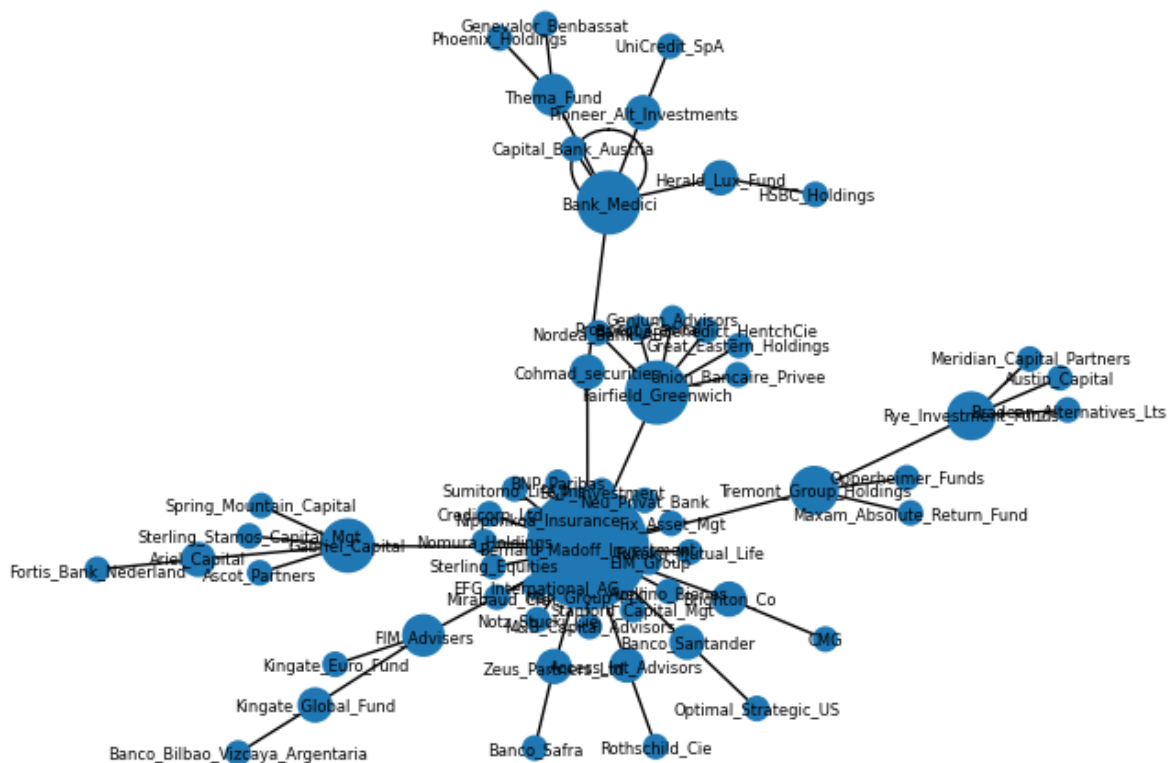
sorted_degree = sorted(deg centrality.items(), key=lambda x: x[1], reverse=True)

print("Top 5 hubs (centralité de degré) :", sorted_degree[:5])

Top 5 hubs (centralité de degré) : [('Bernard_Madoff_Investment', 0.45), ('Bank_Medici', 0.11666666666666667),

sizes = [5000 * deg centrality[n] for n in G.nodes()]

pos = nx.spring_layout(G)
nx.draw(G, pos, with_labels=True, node_size=sizes, font_size=6)
plt.show()
```



```
print("Le réseau est-il connecté :", nx.is_connected(G))
```

```
Le réseau est-il connecté : True
```

```
print("Nombre de composantes connectées :", nx.number_connected_components(G))
```

```
Nombre de composantes connectées : 1
```

```
largest_cc = max(nx.connected_components(G), key=len)
```

```
print("Taille de la plus grande composante :", len(largest_cc))
```

```
Taille de la plus grande composante : 61
```

```
G_giant = G.subgraph(largest_cc)
```

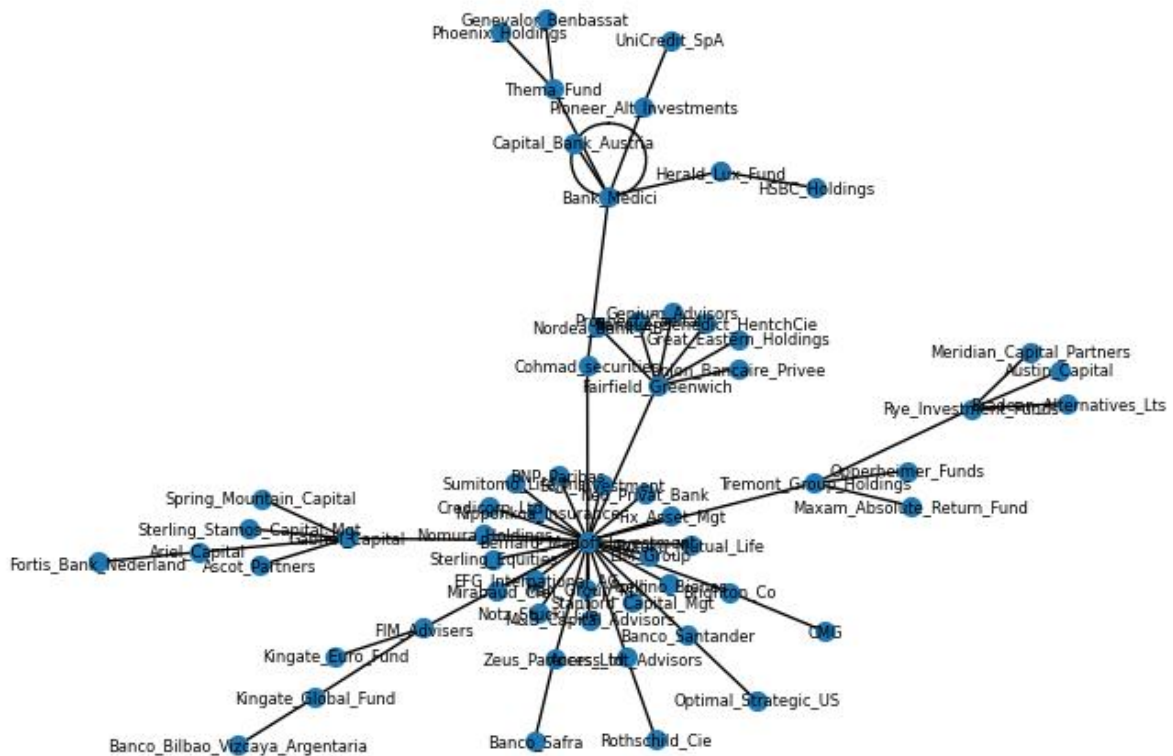
```
plt.figure(figsize=(12,8))
```

```
<Figure size 1200x800 with 0 Axes>
```

```
<Figure size 1200x800 with 0 Axes>
```

```
nx.draw(G_giant, pos, with_labels=True, node_size=50, font_size=6)  
plt.title('Composante géante du réseau Madoff')  
plt.show()
```

Composante géante du réseau Madoff



```
print("Longueur moyenne des chemins (APL) :", nx.average_shortest_path_length(G_giant))
```

Longueur moyenne des chemins (APL) : 3.430601092896175

```
print("Diamètre :", nx.diameter(G_giant))
```

Diamètre : 7

```
print("Coefficient de clustering moyen :", nx.average_clustering(G))
```

Coefficient de clustering moyen : 0.0

```
triangles = sum(nx.triangles(G).values()) // 3
```

```
print("Nombre de triangles dans le réseau :", triangles)
```

Nombre de triangles dans le réseau : 0

```
import networkx.algorithms.community as nxcom

communities = nxcom.greedy_modularity_communities(G)

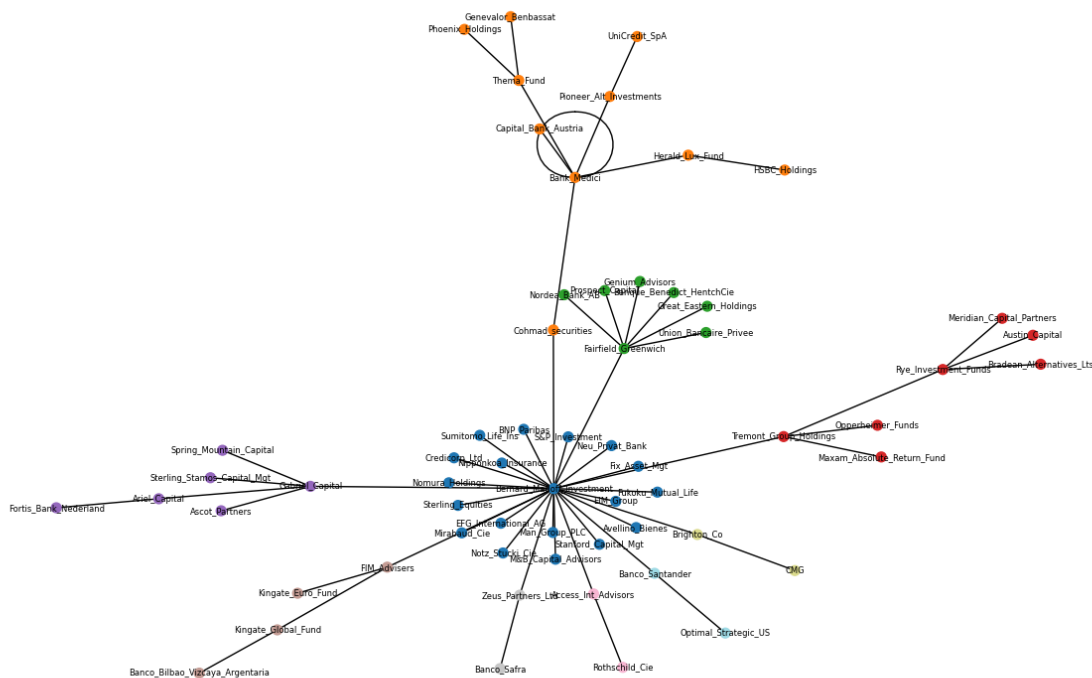
print("Nombre de communautés trouvées :", len(communities))

Nombre de communautés trouvées : 10
```

```
color_map = {}
for c_idx, community in enumerate(communities):
    for name in community:
        color_map[name] = c_idx

colors = [color_map[n] for n in G.nodes()]
plt.figure(figsize=(12,8))
nx.draw(G, pos, node_color=colors, with_labels=True, cmap=plt.cm.tab20, node_size=50, font_size=6)
plt.title('Réseau Madoff - Communautés détectées')
plt.show()
```

Réseau Madoff - Communautés détectées



```
clustering = nx.clustering(G)
sorted_clustering = sorted(clustering.items(), key=lambda x: x[1], reverse=True)
print("Top 5 noeuds avec le plus fort clustering :", sorted_clustering[:5])

Top 5 noeuds avec le plus fort clustering : [('HSBC_Holdings', 0), ('Genevalor_Benbassat', 0), ('Phoenix_Holdings', 0), ('Thema_Fund', 0), ('Herald_Lux_Fund', 0)]
```



```

edge_betweenness = nx.edge_betweenness_centrality(G)
sorted_edges = sorted(edge_betweenness.items(), key=lambda x: x[1], reverse=True)
print("Top 5 liens avec la plus forte intermédiation :", sorted_edges[:5])

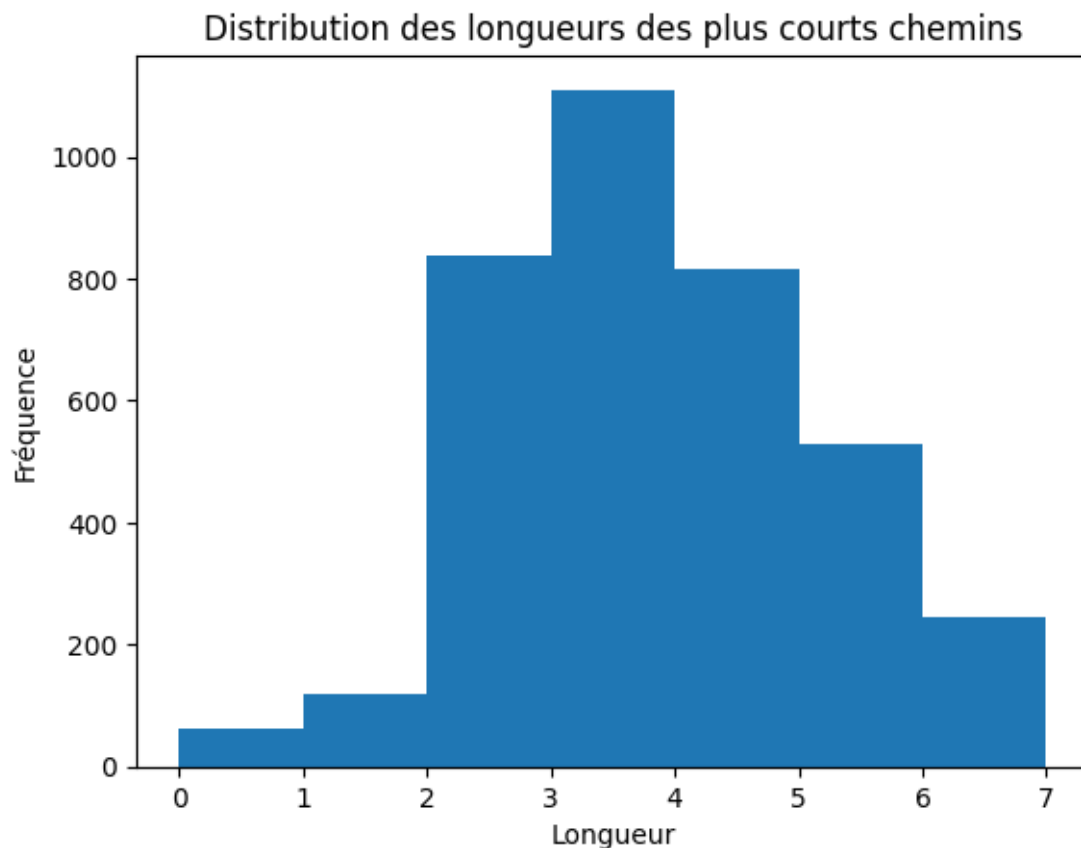
```

Top 5 liens avec la plus forte intermédiation : [(['Cohmad_securities', 'Bernard_Madoff_Investment'), 0.2786885245901639],

```

path_lengths = dict(nx.all_pairs_shortest_path_length(G_giant))
all_lengths = [length for target_dict in path_lengths.values() for length in target_dict.values()]
plt.hist(all_lengths, bins=range(max(all_lengths)+1))
plt.title("Distribution des longueurs des plus courts chemins")
plt.xlabel("Longueur")
plt.ylabel("Fréquence")
plt.show()

```



```

from copy import deepcopy

def robustness_test(G, removal_fraction=0.1):
    G_copy = deepcopy(G)
    num_remove = int(removal_fraction * G_copy.number_of_nodes())
    nodes_sorted = sorted(G_copy.degree, key=lambda x: x[1], reverse=True)
    to_remove = [n for n, d in nodes_sorted[:num_remove]]
    G_copy.remove_nodes_from(to_remove)
    largest_cc = max(nx.connected_components(G_copy), key=len)
    return len(largest_cc) / G.number_of_nodes()

```

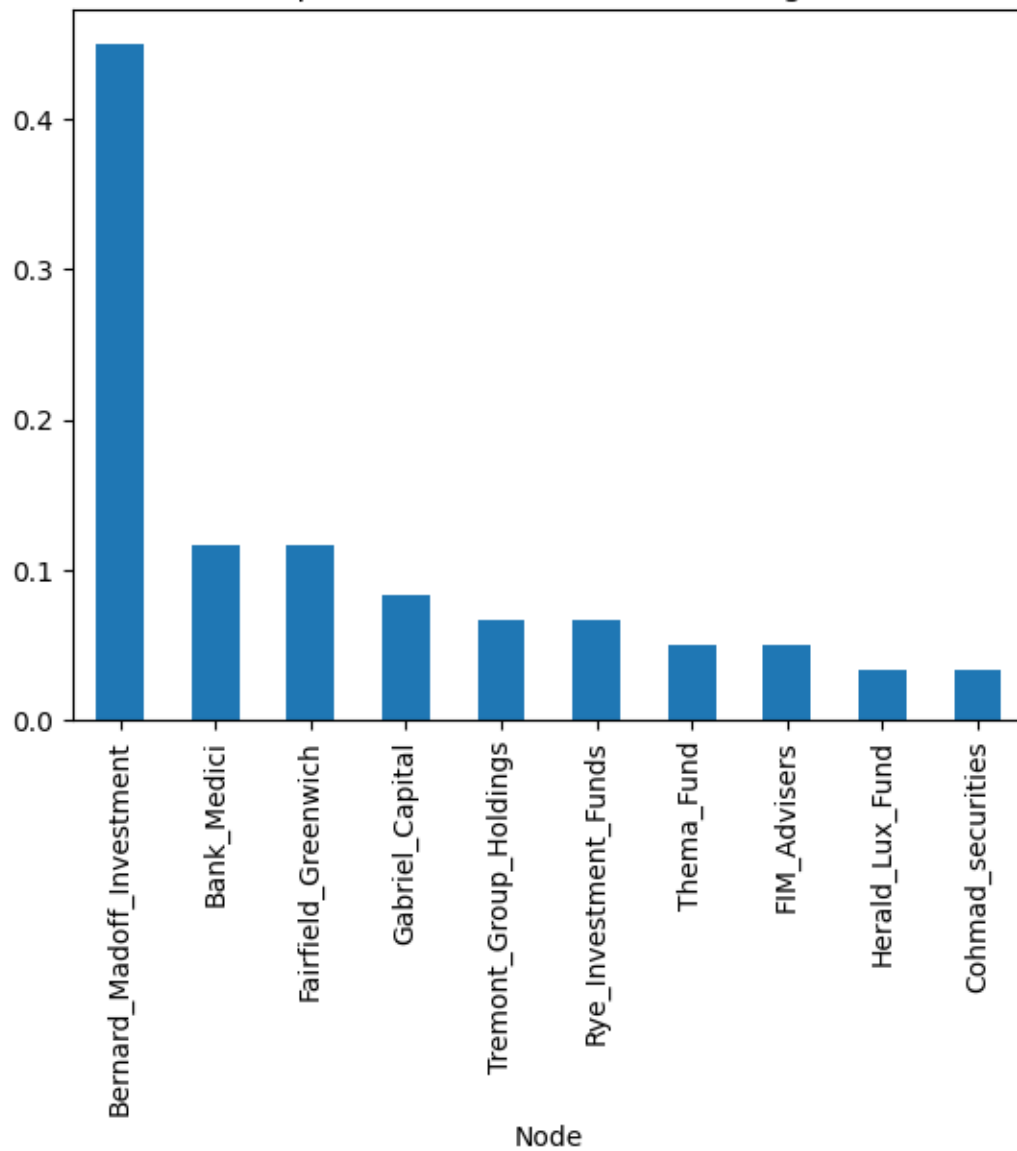
```

print("Taille relative après suppression de 10% des hubs :", robustness_test(G, 0.1))

```

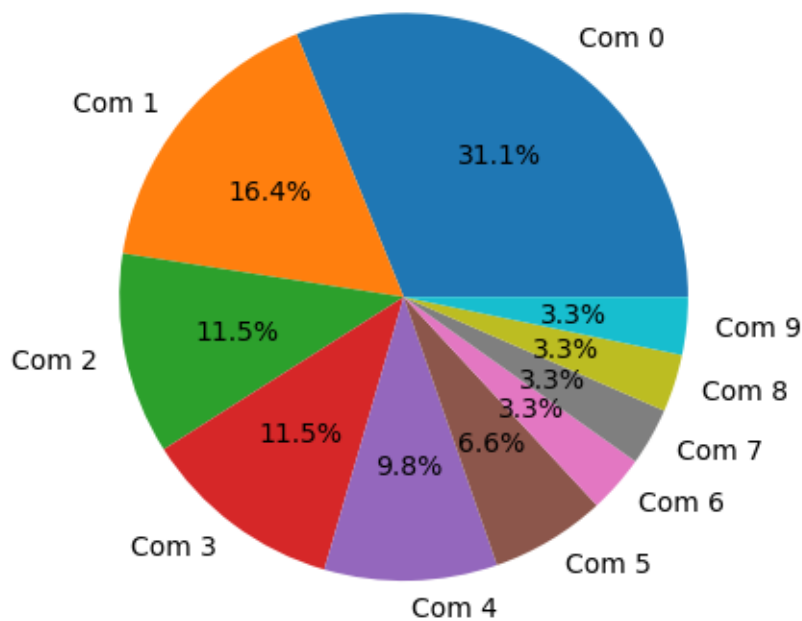
Taille relative après suppression de 10% des hubs : 0.06557377049180328

Top 10 noeuds - Centralité de degré

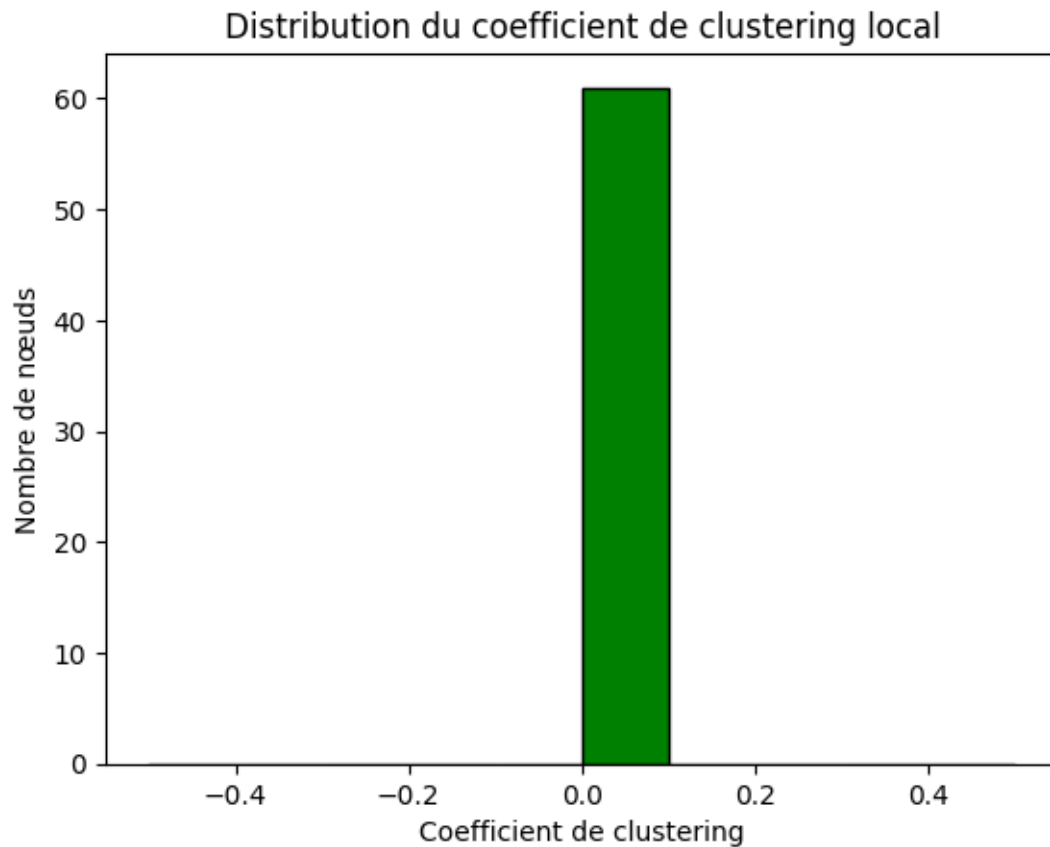


```
community_sizes = [len(c) for c in communities]
plt.pie(community_sizes, labels=[f"Com {i}" for i in range(len(community_sizes))], autopct='%1.1f%%')
plt.title("Répartition des communautés")
plt.show()
```

Répartition des communautés



```
clustering_values = list(clustering.values())
plt.hist(clustering_values, bins=10, color='green', edgecolor='black')
plt.title('Distribution du coefficient de clustering local')
plt.xlabel('Coefficient de clustering')
plt.ylabel('Nombre de nœuds')
plt.show()
```



```

A = nx.adjacency_matrix(G)
print("Matrice d'adjacence :")
print(A.todense())

strength = dict(G.degree(weight='weight'))
print("Force des nœuds :", strength)

A = nx.adjacency_matrix(G)
print("Matrice d'adjacence :")
print(A.todense())

strength = dict(G.degree(weight='weight'))
print("Force des nœuds :", strength)

knn = {node: sum(G.degree(neighbor) for neighbor in G.neighbors(node)) / G.degree(node) if G.degree(node) > 0 else 0 for node in G}
print("Degré moyen des voisins :", knn)

```

```

Matrice d'adjacence :
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
Force des nœuds : {'HSBC_Holdings': 1.0, 'Genevalor_Benbassat': 1.0, 'Phoenix_Holdings': 1.0, 'Thema_Fund': 3.0, 'Herald_Lux_Fund': 2.0, 'Capital_Bank_Austria': 1.0, 'Cohmad_se
Matrice d'adjacence :
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
Force des nœuds : {'HSBC_Holdings': 1.0, 'Genevalor_Benbassat': 1.0, 'Phoenix_Holdings': 1.0, 'Thema_Fund': 3.0, 'Herald_Lux_Fund': 2.0, 'Capital_Bank_Austria': 1.0, 'Cohmad_se
Degré moyen des voisins : {'HSBC_Holdings': 2.0, 'Genevalor_Benbassat': 3.0, 'Phoenix_Holdings': 3.0, 'Thema_Fund': 2.3333333333333335, 'Herald_Lux_Fund': 3.0, 'Capital_Bank_A

```

Bibliographie

M. Fire and C. Guestrin, "The Rise and Fall of Network Stars." Preprint (2017)

Ferretti, S., & D'Angelo, G. (2019). "Dissecting Bitcoin Blockchain: Empirical Analysis of Bitcoin Network". *Journal of Network and Computer Applications*.

Perri, F. S., & Brody, R. G. (2017). "Ponzi of All Ponzis: Critical Analysis of the Bernie Madoff Scheme". *International Journal of Economics and Financial Management*.

Schuchter, A., & Levi, M. (2019). "A Crime Script Analysis of the Madoff Investment Scheme". *Crime, Law and Social Change*.

Lischke, M., & Fabian, B. (2016). "Analyzing the Bitcoin Network: The First Four Years". *Future Internet*.

Investopedia. (2024). "Ponzi Scheme: Definition, Examples, and Origins". <https://www.investopedia.com/terms/p/ponzischeme.asp>

Time. (2021). "Bernie Madoff, Financier Who Operated the Largest Ponzi Scheme in History, Dies". <https://time.com/5788411/bernie-madoff-dies/>

Stanford University. (2014). "Anomaly Detection in the Bitcoin System - A Network Perspective". <https://snap.stanford.edu/class/cs224w-2014/projects2014/cs224w-20-final.pdf>

LawSites. (2009). "The Madoff Social Network Map". <https://www.lawnext.com/2009/11/madoff-social-network-map.html>

Ideal-Investisseur. (2023). "Pyramide de Ponzi : explications de la fraude montée par Madoff". <https://www.ideal-investisseur.fr/economie/comprendre-la-pyramide-de-ponzi-systeme-utilise-par-bernard-madoff-7561.html>

Springer. (2023). "Blockchain Data Analytics from a Network Perspective". In *Blockchain and Distributed Ledger Technology Use Cases*.
https://link.springer.com/chapter/10.1007/978-981-97-4430-5_1