

# Primo Progetto per il corso di Social Computing

## A.A. 2020/21

### Cosa fare

1. Scaricate utenti followers (`api.followers`) e utenti following (`api.friends`) di questi cinque account (i dati numerici potrebbero subire minime variazioni):
  - @mizzaro - 156 Follower - 331 Following
  - @damiano10 - 785 Follower - 836 Following
  - @Miccighel\_ - 331 Follower - 211 Following
  - @eglu81 - 540 Follower - 621 Following
  - @KevinRoitero - 103 Follower - 256 Following
2. Scegliete 5 utenti followers a caso tra quelli di ciascuno dei cinque account e scaricate ulteriori 10 utenti followers (*followers dei followers*)
3. Scegliete 5 utenti following a caso tra quelli di ciascuno dei cinque account e scaricate ulteriori 10 utenti following (*following dei following*)
4. Scaricare i dettagli del profilo di tutti gli utenti recuperati (`api.get_user`)
5. Costruite la rete sociale (grafo):
  - Inserite l'*id* di ciascun utente come identificatore del nodo
  - Ogni arco rappresenta una relazione *follows* tra due utenti
  - La presenza della relazione va verificata tra tutti gli utenti scaricati ed i cinque account iniziali (`api.show_friendship`)
  - Inserite i dettagli del profilo di ciascun utente come *attributi del nodo*
  - Inserite i membri del vostro gruppo come *attributi del grafo*
  - Per ogni nodo, aggiungete un attributo con il numero di follower individuati
6. Producete una visualizzazione interattiva del grafo usando `pyvis`
7. Verificate se il grafo:
  - è connesso (`is_connected`)
  - è bipartito (`is_bipartite`)
8. Misurate le seguenti distanze sul grafo:
  - Centro (`center`)
  - Diametro (`diameter`)
  - Raggio (`radius`)
9. Calcolate le seguenti misure di centralità sul grafo:
  - Betweenness centrality (`betweenness_centrality`)
  - Closeness centrality (`closeness_centrality`)
  - Degree centrality (`degree_centrality`)
  - In-degree centrality (`in_degree_centrality`)
  - Out-degree centrality (`out_degree_centrality`)
  - Page Rank (`pagerank`)
  - HITS (`hits`)
10. Generate il sottografo indotto dal nodo damiano10 (`ego_graph`) e calcolate:
  - Cricca massima (`max_clique`)
  - Dimensione della cricca massima (`large_clique_size`)
11. Calcolate la copertura minima degli archi (`min_edge_cover`) del grafo
12. Calcolate i seguenti coefficienti per stimare la "small-world-ness" del grafo:
  - Coefficiente omega (`omega`)
  - Coefficiente sigma (`sigma`)
13. Calcolare la correlazione di Pearson Rho e di Kendall Tau fra le misure di centralità; riportare il risultato in due tabelle

## Informazioni aggiuntive

- Viste le limitazioni poste sugli endpoint da Twitter, durante la prima fase parallelizzate il lavoro (ognuno scarica dati da un account alla volta), serializzate e deserializzate a posteriori
- Tutte le misure, proprietà e verifiche richieste sono definite nella documentazione di NetworkX (i link alla documentazione di NetworkX vengono riportati accanto al nome di ciascuna misura richiesta)
- **Bisogna usare le funzionalità specificate nel testo del progetto, quando rese esplicite (ad esempio, scaricare i followers di un account utilizzando `api.followers`)**
- **Tuttavia, se trovate soluzioni alternative e/o migliori potete proporle *in aggiunta* a quanto richiesto, dicendolo esplicitamente nella relazione**

## Come consegnare

1. I gruppi **devono** essere formati da **quattro** persone (i gruppi più o meno numerosi verranno penalizzati)
2. Si devono consegnare i seguenti file:
  - Relazione di al massimo 5 pagine (con anche i vostri nomi cognomi e numeri di matricola) che descrive tutto il lavoro svolto
  - Una serializzazione dei dati scaricati mediante API di Twitter con le funzioni spiegate a lezione
  - Una serializzazione del grafo prodotta con le funzioni spiegate a lezione
  - Il codice prodotto (in un unico notebook)
  - Un file `.html` con la visualizzazione interattiva del grafo prodotta mediante `pyvis`
3. Consegnare via mail a entrambi i docenti (un unico messaggio indirizzato a entrambi)
  - [mizzaro@uniud.it](mailto:mizzaro@uniud.it)
  - [michael.soprano@uniud.it](mailto:michael.soprano@uniud.it)
  - oggetto della mail nel formato:  
[Progetto SocCom 1] cognome1\_cognome2\_cognome3\_cognome4
  - in allegato alla mail un unico file zippato che quando scompattato produce una singola cartella con nome cognome1\_cognome2\_cognome3\_cognome4
4. **Scadenza: Lunedì 30 Novembre 2020 AoE Timezone**
5. Punteggio:
  - 5 punti in trentesimi per i migliori 20%,
  - 4 punti per i seguenti 20%,
  - 3 punti per i seguenti 20%,
  - 2 punti seguenti 20%,
  - 1 punto per i seguenti 20%,
  - 0 punti a discrezione dei docenti per progetti non adeguati o per chi non consegna