Breadth-First Search: given some starting vertex $s$, find vertex 1 edge away from $s$.
find vertex 2 edges away.

BFS tree:

- use a queue
- initially queue has starting vertex
- add vertex at distance 1
- add vertex at distance 2

WHITE: haven't seen yet
GRAY: in queue
BLACK: done with

pred (B) = A
pred (C) = B
BFS(V,E,s) — finds shortest paths in unweighted graph

for each vertex u in V
  color[u] <- WHITE
  d[u] <- infinity
  pred[u] <- NIL

F <- []

color[s] <- GRAY
d[s] <- 0
pred[s] <- NIL
Q <- [s]

while not Q.isEmpty() -> loops over all vertices reachable from s ⊆ O(V)
  u <- Q.dequeue()
  for each v adjacent to u
    if color[v] = WHITE
      pred[v] <- u
      d[v] <- d[u] + 1
      color[v] <- GRAY
      Q.enqueue(v)
  color[u] = BLACK
  F = F + u

O(V+E)  O(V^2)

\[
\sum_{u \in V} \sum_{v \in Adj[u]} \frac{1}{\text{degree}(u)} \sum_{i=1}^{\text{degree}(u)} \sum_{j=i+1}^{\eta} \frac{1}{\text{degree}(u)}
\]