package com.twitter.recos.user\_tweet\_graph.util

import com.twitter.graphjet.algorithms.TweetIDMask

import com.twitter.graphjet.bipartite.api.BipartiteGraph

object GetAllInternalTweetIdsUtil {

def getAllInternalTweetIds(tweetId: Long, bipartiteGraph: BipartiteGraph): Seq[Long] = {

val internalTweetIds = getAllMasks(tweetId)

sortByDegrees(internalTweetIds, bipartiteGraph)

}

private def getAllMasks(tweetId: Long): Seq[Long] = {

Seq(

tweetId,

TweetIDMask.summary(tweetId),

TweetIDMask.photo(tweetId),

TweetIDMask.player(tweetId),

TweetIDMask.promotion(tweetId)

)

}

private def sortByDegrees(

encodedTweetIds: Seq[Long],

bipartiteGraph: BipartiteGraph

): Seq[Long] = {

encodedTweetIds

.map { encodedTweetId => (encodedTweetId, bipartiteGraph.getRightNodeDegree(encodedTweetId)) }

.filter { case (\_, degree) => degree > 0 } // keep only tweetds with positive degree

.sortBy { case (\_, degree) => -degree } // sort by degree in descending order

.map { case (encodedTweetId, \_) => encodedTweetId }

}

}