**Information Retrieval**

**16 December 2019 (FinalTerm)**

**Name: Surname: Matricola:**

**Ex 1 [ranks 3+3+2]** Given the following four texts:

* + T1=”a beautiful toy”
	+ T2=”toy one and toy two”
	+ T3=”a new toy”
	+ T4=”one two.... toy”
1. Compute the TF-IDF vectors of the four texts above (logs are in base two).
2. Find the most similar text to T3 in the vector space model (no normalization).
3. Find the most similar text to the query q = [one one toy] in the vector space model (no normalization).

**Ex 2 [points 4]** Consider the WAND algorithm over the following four posting lists by assuming that at some step the scanning algorithm is examining the first showed number of each list

t1 🡪 (…, 5, 6, 7, 8, 15)

t2 🡪 (…, 1, 3, 4, 5, 7, 8, 14)

t3 🡪 (…, 8, 13, 15)

t4 🡪 (…, 3, 4, 7, 8, 9)

At that time assume that the current threshold equals 3.1, and the upper bounds of the scores in each posting list are: ub\_1 = 0.2, ub\_2 = 2, ub\_3 = 4, ub\_4 = 0.3.

Which is the next docID whose full score is computed? *(Motivate your answer)*

**Ex 3 [points 4+4+2]** Given the graph



* Execute one step of the PageRank algorithm, assuming uniform teleportation step, alpha = ½ and uniform starting probability.
* Execute one step of the Personalized PageRank algorithm with respect to the node 2, assuming alpha = ½ and uniform starting probability.
* Which is the node most similar to node 2?

**Ex 4 [points 4+4]** Answer the following two questions:

* Define mathematically the two features *link probability* and *commonness,* and comment their significance in their usage in Entity Linkers (such as TagMe)
* Show how TF-IDF is “stored” within the posting lists to allow its retrieval when query its executed