Input Parameters

- Original ranking, `existingResults`
- Diversification depth, set in this study to `k=30`
- `w(=0.7)`, weighting for diversification scoring component

Output

- Manipulated array of results, diversified to depth `k` (see above)

Helper Functions

- `getEntities(x,y,z)` Given an array of results (documents), returns an array of entities present in the results array from range `y` to `z`
- `getLength(x)` Returns the length of array `x`
- `getUnseenEntities(x,y)` Returns entities in document `x` that have not yet been observed in ranked document array `y`
- `sortByScore(x)` Sorts document array `x` by `score` in descending order
- `<array>.pop()` Removes the top entry from an array, returning the popped value

```
SET entities TO []
SET newRankings TO []
SET i TO 1

# Take the top result from the baseline results, popping results
SET newRankings[0] TO existingResults.pop()

WHILE i <= k DO
    # Obtain all entities from the first to ith result
    SET entities TO getEntities(existingResults, 0, i-1)
    SET j TO 0

    # Now rescore all remaining results, considering weighting w
    WHILE j <= getLength(existingResults) DO
        SET newEntityCount TO
                getUnseenEntities(document, existingResults)
        SET existingResults[j].score TO score + (w·newEntityCount)
        SET j TO j + 1
    END WHILE

    # Reorder existingResults; move top result to new array
    sortByScore(existingResults)
    SET newRankings[i] TO existingResults.pop()
    SET i TO i + 1
END WHILE
```