Homework 3

Ilai Fallach, 200924751 and Orr Mandelbaum 200612190

Question 1

Pseudo code for K-Means in MapReduce paradigm:

```
map_fn(k, centers, point):
    new_k = find_nearest_center(centers, point)
    return new_k, point
end

reduce_fn(k, points):
    new_center = find_new_center(points)
    cost = compute_cost(new_center, points)
    return new_center, points, center
end

main():
    step_delta_threshold = 1
    points = load_points()
    centers = initialize_random_centers(points)

    step_delta = step_delta_threshold + 1
    last_cost = None

    while step_delta > step_delta_threshold:
        centers, points, cost = \
            run_map_reduce_job(centers, points, map_fn, reduce_fn)
        if last_cost is not None:
            step_delta = last_cost - cost
        last_cost = cost
    end

    return centers, points
end
```

Question 2

Pseudo code for CheckClique in MapReduce paradigm:

```
map_fn(k, v):
        yield 1, 1
end

map_fn_2(k, v):
        d = v.split("->")
        neighbours = d[1].split(" ")
        yield len(neighbours), 1
end

reduce_fn(k, v):
        return sum(v)
end

main():
        graph_text = load_graph()

        num_vertices = run_map_reduce_job(graph_text, map_fn, reduce_fn)
        result = run_map_reduce_job(graph_text, map_fn_2, reduce_fn)

        if num_vertices – 1 is in result and \
                result[num_vertices – 1] == num_vertices:
                return True
        else
                return False
        end
end
```

Question 3

Pseudo code for Pseudo-Synonyms Detection in MapReduce paradigm:

```
map_fn(k, v):
        words = v.split(" ")
        yield (words[0], words[2]), words[1]
end

reduce_fn(k, v):
```

```
        pairs = compute_pairwise_combinations(v)
        return pairs
end

map_fn_2(k, v):
        first, second = v
        if first > second:
                second, first = first, second
        yield (first, second), 1
end

reduce_fn_2(k, v):
        return sum(v)
end

main():
        text = load_text()

        pairs = run_map_reduce_job(text, map_fn, reduce_fn)
        pairs_count = run_map_reduce_job(pairs, map_fn_2, reduce_fn_2)

        return pairs_count
end
```