CSCI 301
Lab 7 (Operations on Bags)
25 Points Total

Bags are structures very like sets save that they permit multiple instances as members. Like sets, order is irrelevant. For example, the following bag contents are permitted:

$$[a, a, a, b, b, b, b]$$

and

$$[a, a, a, a, b, b, b]$$

Because of this difference between bags and sets, the bag-theoretic operations are somewhat different from those defined for sets. Here are three of them.

1. Bag-Difference. The operation Bag-Difference when applied to two bags results in a bag where each element appears as many times as it appears in the first bag, minus the number of times it appears in the second bag (but never less than 0 times). For example

```
(bag-difference ’(a a b a) ’(b a a)) --->’(a)
(bag-difference ’(a b a a) ’(a b c)) --->’(a a)
(bag-difference ’(a b c) ’(a b a a)) --->’(c)
(bag-difference ’(a b c) ’(a b c)) --->’()
(bag-difference ’() ’(a b a a)) --->’()
(bag-difference ’(a b a a) ’())--->’(b a a a)
```

2. Bag-Union. The operation Bag-Union results in a bag that contains the maximum number elements that are contained in the operands. For example

```
(bag-union ’(a a b a) ’(b a a)) --->’(a b a a)
(bag-union ’(a b a a) ’(a b c)) --->’(a a a b c)
(bag-union ’(a b c) ’(a b a a)) --->’(c a b a a)
(bag-union ’(a b c) ’(a b c)) --->’(a b c)
(bag-union ’() ’(a b a a)) --->’(a b a a)
(bag-union ’(a b a a) ’())--->’(a b a a)
```

3. Bag-Intersection. The operation Bag-Intersection results in a bag that contains the minimum number of elements that are contained in the bag operands. For example

```
(bag-intersection ’(a a b a) ’(b a a)) ---> ’(b a a)
(bag-intersection ’(a b a a) ’(a b c)) ---> ’(b a)
(bag-intersection ’(a b c) ’(a b a a)) --->’(a b)
(bag-intersection ’(a b c) ’(a b c)) --->’(a b c)
(bag-intersection ’() ’(a b a a)) --->’()
(bag-intersection ’(a b a a) ’())--->’()
```

The solutions will be turned in by posting a single Racket program (lab07. rkt) containing a definition of all the functions specified.