X5: Restriction Mapping Algorithms

Please submit your answers (as an *.rkt* file) via e-mail within a week (Subject should be: DCB X5). Keep in mind that the file must run without errors, and any procedures (names, arguments etc.) specified in a given task must be maintained.

Note: for each successfully completed lab-report you get a BONUS point that will be added on the final written examination points you will score.

Task 1

You were shown a Scheme implementation of the restriction mapping algorithm called *PartialDigest* at the lecture. Here you get an incomplete implementation of the *BruteForcePDP* algorithm described at the same lecture, missing the procedure **multiset**.

A multiset ΔX is the set of all pairwise positive distances between elements in a set X, e.g. the multiset of X = {0, 2, 4, 7, 10} is ΔX = {2, 2, 3, 3, 4, 5, 6, 7, 8, 10}.

a) Write a Scheme procedure **multiset**, that given the set X, returns the multiset ΔX. You may use the Δ(y, X) procedure implemented as **delta** in the *PartialDigest* algorithm.
b) What is the time complexity of **multiset**?

Task 2

Consider the partial digest L= {1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 5, 6, 6, 6, 9, 9, 10, 11, 12, 15}.

Follow the *PartialDigest* algorithm below by hand to solve the partial digest problem for L (i.e. find X such that ΔX = L). Illustrate the recursive calls by drawing a tree.

You may use the Scheme implementation and the debugger to verify your solution.

```
PartialDigest(L)
1     width ← Maximum element in L
2     Remove width from  L
3     X ← {0, width}
4     Place(L, X)

Place(L,X)
1     if L is empty
2          output X
3          return
4     y ← Maximum element in L
5     if Δ(y, X )  L
```

```
6          Remove lengths Δ(y, X) from L and add y to X
7          Place(L, X )
8          Remove y from X and add lengths Δ(y, X) to L
9     if Δ(width – y, X )  L
10         Remove lengths Δ(width – y, X) from L and add width – y to
X and
11         Place(L, X )
12         Remove width – y from X and add lengths Δ(width – y, X ) to
L
13    return
```