IS622 - Homework 7

Brian Chu | Oct 10, 2015

Exercise 6.1.1 : Suppose there are 100 items, numbered 1 to 100, and also 100 baskets, also numbered 1 to 100. Item i is in basket b if and only if i divides b with no remainder. Thus, item 1 is in all the baskets, item 2 is in all fifty of the even-numbered baskets, and so on. Basket 12 consists of items {1, 2, 3, 4, 6, 12}, since these are all the integers that divide 12. Answer the following questions:

(a) If the support threshold is 5, which items are frequent?

```
support <- 5
frequent <- character()
total_basketcount <- 0

# Loop through baskets
for (i in 1:100) {
  basketcount <- 0
  item <- character()

  # Loop through items
  for (j in 1:100) {
    if (j %% i == 0) {
      basketcount <- basketcount + 1
      item <- c(item, j)
    }
  }
  item <- paste(item, collapse=" ")

  # For tracking part C
  total_basketcount <- total_basketcount + basketcount

  # Count only frequent items
  if (basketcount >=5) {
    frequent <- c(frequent, i)
  }
  print(paste("item:", i, "| frequency:", basketcount, "| baskets:", item))
}
```

```
## [1] "item: 1 | frequency: 100 | baskets: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [1] "item: 2 | frequency: 50 | baskets: 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 4
## [1] "item: 3 | frequency: 33 | baskets: 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 
## [1] "item: 4 | frequency: 25 | baskets: 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84
## [1] "item: 5 | frequency: 20 | baskets: 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100"
## [1] "item: 6 | frequency: 16 | baskets: 6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96"
## [1] "item: 7 | frequency: 14 | baskets: 7 14 21 28 35 42 49 56 63 70 77 84 91 98"
## [1] "item: 8 | frequency: 12 | baskets: 8 16 24 32 40 48 56 64 72 80 88 96"
## [1] "item: 9 | frequency: 11 | baskets: 9 18 27 36 45 54 63 72 81 90 99"
## [1] "item: 10 | frequency: 10 | baskets: 10 20 30 40 50 60 70 80 90 100"
## [1] "item: 11 | frequency: 9 | baskets: 11 22 33 44 55 66 77 88 99"
## [1] "item: 12 | frequency: 8 | baskets: 12 24 36 48 60 72 84 96"
## [1] "item: 13 | frequency: 7 | baskets: 13 26 39 52 65 78 91"
```

```
## [1] "item: 14 | frequency: 7 | baskets: 14 28 42 56 70 84 98"
## [1] "item: 15 | frequency: 6 | baskets: 15 30 45 60 75 90"
## [1] "item: 16 | frequency: 6 | baskets: 16 32 48 64 80 96"
## [1] "item: 17 | frequency: 5 | baskets: 17 34 51 68 85"
## [1] "item: 18 | frequency: 5 | baskets: 18 36 54 72 90"
## [1] "item: 19 | frequency: 5 | baskets: 19 38 57 76 95"
## [1] "item: 20 | frequency: 5 | baskets: 20 40 60 80 100"
## [1] "item: 21 | frequency: 4 | baskets: 21 42 63 84"
## [1] "item: 22 | frequency: 4 | baskets: 22 44 66 88"
## [1] "item: 23 | frequency: 4 | baskets: 23 46 69 92"
## [1] "item: 24 | frequency: 4 | baskets: 24 48 72 96"
## [1] "item: 25 | frequency: 4 | baskets: 25 50 75 100"
## [1] "item: 26 | frequency: 3 | baskets: 26 52 78"
## [1] "item: 27 | frequency: 3 | baskets: 27 54 81"
## [1] "item: 28 | frequency: 3 | baskets: 28 56 84"
## [1] "item: 29 | frequency: 3 | baskets: 29 58 87"
## [1] "item: 30 | frequency: 3 | baskets: 30 60 90"
## [1] "item: 31 | frequency: 3 | baskets: 31 62 93"
## [1] "item: 32 | frequency: 3 | baskets: 32 64 96"
## [1] "item: 33 | frequency: 3 | baskets: 33 66 99"
## [1] "item: 34 | frequency: 2 | baskets: 34 68"
## [1] "item: 35 | frequency: 2 | baskets: 35 70"
## [1] "item: 36 | frequency: 2 | baskets: 36 72"
## [1] "item: 37 | frequency: 2 | baskets: 37 74"
## [1] "item: 38 | frequency: 2 | baskets: 38 76"
## [1] "item: 39 | frequency: 2 | baskets: 39 78"
## [1] "item: 40 | frequency: 2 | baskets: 40 80"
## [1] "item: 41 | frequency: 2 | baskets: 41 82"
## [1] "item: 42 | frequency: 2 | baskets: 42 84"
## [1] "item: 43 | frequency: 2 | baskets: 43 86"
## [1] "item: 44 | frequency: 2 | baskets: 44 88"
## [1] "item: 45 | frequency: 2 | baskets: 45 90"
## [1] "item: 46 | frequency: 2 | baskets: 46 92"
## [1] "item: 47 | frequency: 2 | baskets: 47 94"
## [1] "item: 48 | frequency: 2 | baskets: 48 96"
## [1] "item: 49 | frequency: 2 | baskets: 49 98"
## [1] "item: 50 | frequency: 2 | baskets: 50 100"
## [1] "item: 51 | frequency: 1 | baskets: 51"
## [1] "item: 52 | frequency: 1 | baskets: 52"
## [1] "item: 53 | frequency: 1 | baskets: 53"
## [1] "item: 54 | frequency: 1 | baskets: 54"
## [1] "item: 55 | frequency: 1 | baskets: 55"
## [1] "item: 56 | frequency: 1 | baskets: 56"
## [1] "item: 57 | frequency: 1 | baskets: 57"
## [1] "item: 58 | frequency: 1 | baskets: 58"
## [1] "item: 59 | frequency: 1 | baskets: 59"
## [1] "item: 60 | frequency: 1 | baskets: 60"
## [1] "item: 61 | frequency: 1 | baskets: 61"
## [1] "item: 62 | frequency: 1 | baskets: 62"
## [1] "item: 63 | frequency: 1 | baskets: 63"
## [1] "item: 64 | frequency: 1 | baskets: 64"
## [1] "item: 65 | frequency: 1 | baskets: 65"
## [1] "item: 66 | frequency: 1 | baskets: 66"
## [1] "item: 67 | frequency: 1 | baskets: 67"
```

```
## [1] "item: 68 | frequency: 1 | baskets: 68"
## [1] "item: 69 | frequency: 1 | baskets: 69"
## [1] "item: 70 | frequency: 1 | baskets: 70"
## [1] "item: 71 | frequency: 1 | baskets: 71"
## [1] "item: 72 | frequency: 1 | baskets: 72"
## [1] "item: 73 | frequency: 1 | baskets: 73"
## [1] "item: 74 | frequency: 1 | baskets: 74"
## [1] "item: 75 | frequency: 1 | baskets: 75"
## [1] "item: 76 | frequency: 1 | baskets: 76"
## [1] "item: 77 | frequency: 1 | baskets: 77"
## [1] "item: 78 | frequency: 1 | baskets: 78"
## [1] "item: 79 | frequency: 1 | baskets: 79"
## [1] "item: 80 | frequency: 1 | baskets: 80"
## [1] "item: 81 | frequency: 1 | baskets: 81"
## [1] "item: 82 | frequency: 1 | baskets: 82"
## [1] "item: 83 | frequency: 1 | baskets: 83"
## [1] "item: 84 | frequency: 1 | baskets: 84"
## [1] "item: 85 | frequency: 1 | baskets: 85"
## [1] "item: 86 | frequency: 1 | baskets: 86"
## [1] "item: 87 | frequency: 1 | baskets: 87"
## [1] "item: 88 | frequency: 1 | baskets: 88"
## [1] "item: 89 | frequency: 1 | baskets: 89"
## [1] "item: 90 | frequency: 1 | baskets: 90"
## [1] "item: 91 | frequency: 1 | baskets: 91"
## [1] "item: 92 | frequency: 1 | baskets: 92"
## [1] "item: 93 | frequency: 1 | baskets: 93"
## [1] "item: 94 | frequency: 1 | baskets: 94"
## [1] "item: 95 | frequency: 1 | baskets: 95"
## [1] "item: 96 | frequency: 1 | baskets: 96"
## [1] "item: 97 | frequency: 1 | baskets: 97"
## [1] "item: 98 | frequency: 1 | baskets: 98"
## [1] "item: 99 | frequency: 1 | baskets: 99"
## [1] "item: 100 | frequency: 1 | baskets: 100"
```

```
length(frequent)
```

```
## [1] 20
```

```
print(frequent)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14"
## [15] "15" "16" "17" "18" "19" "20"
```

Each of items 1-20 appear in at least 5 baskets. This is also intuitive since $100/20 = 5$.

(b) If the support threshold is 5, which pairs of items are frequent?

```
# Create list of possible pairs
f <- expand.grid(1:20, 1:20)
```

```r
f2 <- f[f[,2] > f[,1],]
f2 <- f2[order(f2[,1]),]

freqcount <- 0

# Loop through pairs
for (i in 1:nrow(f2)) {
  doublecount <- 0
  # Loop through baskets
  for (j in 1:100) {
    if ((j %% f2[i,1] == 0) & (j %% f2[i,2] == 0)) {
      doublecount <- doublecount + 1
    }
  }
  # Count if pairs in at least 5 baskets, increment total count of frequent baskets
  if (doublecount >= 5) {
    freqpair <- paste("(", f2[i,1], ",", f2[i,2], ")", sep="")
    print(paste("pair:",freqpair, "| frequency:",doublecount))
    freqcount <- freqcount + 1
  }

}
```

```
## [1] "pair: (1,2) | frequency: 50"
## [1] "pair: (1,3) | frequency: 33"
## [1] "pair: (1,4) | frequency: 25"
## [1] "pair: (1,5) | frequency: 20"
## [1] "pair: (1,6) | frequency: 16"
## [1] "pair: (1,7) | frequency: 14"
## [1] "pair: (1,8) | frequency: 12"
## [1] "pair: (1,9) | frequency: 11"
## [1] "pair: (1,10) | frequency: 10"
## [1] "pair: (1,11) | frequency: 9"
## [1] "pair: (1,12) | frequency: 8"
## [1] "pair: (1,13) | frequency: 7"
## [1] "pair: (1,14) | frequency: 7"
## [1] "pair: (1,15) | frequency: 6"
## [1] "pair: (1,16) | frequency: 6"
## [1] "pair: (1,17) | frequency: 5"
## [1] "pair: (1,18) | frequency: 5"
## [1] "pair: (1,19) | frequency: 5"
## [1] "pair: (1,20) | frequency: 5"
## [1] "pair: (2,3) | frequency: 16"
## [1] "pair: (2,4) | frequency: 25"
## [1] "pair: (2,5) | frequency: 10"
## [1] "pair: (2,6) | frequency: 16"
## [1] "pair: (2,7) | frequency: 7"
## [1] "pair: (2,8) | frequency: 12"
## [1] "pair: (2,9) | frequency: 5"
## [1] "pair: (2,10) | frequency: 10"
## [1] "pair: (2,12) | frequency: 8"
## [1] "pair: (2,14) | frequency: 7"
## [1] "pair: (2,16) | frequency: 6"
```

```
## [1] "pair: (2,18) | frequency: 5"
## [1] "pair: (2,20) | frequency: 5"
## [1] "pair: (3,4) | frequency: 8"
## [1] "pair: (3,5) | frequency: 6"
## [1] "pair: (3,6) | frequency: 16"
## [1] "pair: (3,9) | frequency: 11"
## [1] "pair: (3,12) | frequency: 8"
## [1] "pair: (3,15) | frequency: 6"
## [1] "pair: (3,18) | frequency: 5"
## [1] "pair: (4,5) | frequency: 5"
## [1] "pair: (4,6) | frequency: 8"
## [1] "pair: (4,8) | frequency: 12"
## [1] "pair: (4,10) | frequency: 5"
## [1] "pair: (4,12) | frequency: 8"
## [1] "pair: (4,16) | frequency: 6"
## [1] "pair: (4,20) | frequency: 5"
## [1] "pair: (5,10) | frequency: 10"
## [1] "pair: (5,15) | frequency: 6"
## [1] "pair: (5,20) | frequency: 5"
## [1] "pair: (6,9) | frequency: 5"
## [1] "pair: (6,12) | frequency: 8"
## [1] "pair: (6,18) | frequency: 5"
## [1] "pair: (7,14) | frequency: 7"
## [1] "pair: (8,16) | frequency: 6"
## [1] "pair: (9,18) | frequency: 5"
## [1] "pair: (10,20) | frequency: 5"
```

```r
print(freqcount)
```

```
## [1] 56
```

There are 56 pairs that are in at least 5 baskets together.

(c) What is the sum of the sizes of all the baskets?

```r
# Calculated in Part A code
print(total_basketcount)
```

```
## [1] 482
```

There are 482 total items in the 100 baskets (i.e. sum of all sizes)

I also computed the items in each basket below. Summing up each basket also results in 482 total items.

```r
# Items in each basket, count which ones have at least 5 items
# Did this additionally, but not necessary to answer question
support <- 5
abovesupport <- character()
total_itemcount <- 0
```

```
for (i in 1:100) {
  itemcount <- 0
  basket <- character()

  for (j in 1:100) {
    if (i %% j == 0) {
      itemcount <- itemcount + 1
      basket <- c(basket, j)
    }
  }
  basket <- paste(basket, collapse=" ")

  total_itemcount <- total_itemcount + itemcount

  if (itemcount >=5) {
    abovesupport <- c(abovesupport, i)
  }
  print(paste("basket:", i, "| frequency:", itemcount, "| items:", basket))
}
```

```
## [1] "basket: 1 | frequency: 1 | items: 1"
## [1] "basket: 2 | frequency: 2 | items: 1 2"
## [1] "basket: 3 | frequency: 2 | items: 1 3"
## [1] "basket: 4 | frequency: 3 | items: 1 2 4"
## [1] "basket: 5 | frequency: 2 | items: 1 5"
## [1] "basket: 6 | frequency: 4 | items: 1 2 3 6"
## [1] "basket: 7 | frequency: 2 | items: 1 7"
## [1] "basket: 8 | frequency: 4 | items: 1 2 4 8"
## [1] "basket: 9 | frequency: 3 | items: 1 3 9"
## [1] "basket: 10 | frequency: 4 | items: 1 2 5 10"
## [1] "basket: 11 | frequency: 2 | items: 1 11"
## [1] "basket: 12 | frequency: 6 | items: 1 2 3 4 6 12"
## [1] "basket: 13 | frequency: 2 | items: 1 13"
## [1] "basket: 14 | frequency: 4 | items: 1 2 7 14"
## [1] "basket: 15 | frequency: 4 | items: 1 3 5 15"
## [1] "basket: 16 | frequency: 5 | items: 1 2 4 8 16"
## [1] "basket: 17 | frequency: 2 | items: 1 17"
## [1] "basket: 18 | frequency: 6 | items: 1 2 3 6 9 18"
## [1] "basket: 19 | frequency: 2 | items: 1 19"
## [1] "basket: 20 | frequency: 6 | items: 1 2 4 5 10 20"
## [1] "basket: 21 | frequency: 4 | items: 1 3 7 21"
## [1] "basket: 22 | frequency: 4 | items: 1 2 11 22"
## [1] "basket: 23 | frequency: 2 | items: 1 23"
## [1] "basket: 24 | frequency: 8 | items: 1 2 3 4 6 8 12 24"
## [1] "basket: 25 | frequency: 3 | items: 1 5 25"
## [1] "basket: 26 | frequency: 4 | items: 1 2 13 26"
## [1] "basket: 27 | frequency: 4 | items: 1 3 9 27"
## [1] "basket: 28 | frequency: 6 | items: 1 2 4 7 14 28"
## [1] "basket: 29 | frequency: 2 | items: 1 29"
## [1] "basket: 30 | frequency: 8 | items: 1 2 3 5 6 10 15 30"
## [1] "basket: 31 | frequency: 2 | items: 1 31"
## [1] "basket: 32 | frequency: 6 | items: 1 2 4 8 16 32"
## [1] "basket: 33 | frequency: 4 | items: 1 3 11 33"
```

```
## [1] "basket: 34 | frequency: 4 | items: 1 2 17 34"
## [1] "basket: 35 | frequency: 4 | items: 1 5 7 35"
## [1] "basket: 36 | frequency: 9 | items: 1 2 3 4 6 9 12 18 36"
## [1] "basket: 37 | frequency: 2 | items: 1 37"
## [1] "basket: 38 | frequency: 4 | items: 1 2 19 38"
## [1] "basket: 39 | frequency: 4 | items: 1 3 13 39"
## [1] "basket: 40 | frequency: 8 | items: 1 2 4 5 8 10 20 40"
## [1] "basket: 41 | frequency: 2 | items: 1 41"
## [1] "basket: 42 | frequency: 8 | items: 1 2 3 6 7 14 21 42"
## [1] "basket: 43 | frequency: 2 | items: 1 43"
## [1] "basket: 44 | frequency: 6 | items: 1 2 4 11 22 44"
## [1] "basket: 45 | frequency: 6 | items: 1 3 5 9 15 45"
## [1] "basket: 46 | frequency: 4 | items: 1 2 23 46"
## [1] "basket: 47 | frequency: 2 | items: 1 47"
## [1] "basket: 48 | frequency: 10 | items: 1 2 3 4 6 8 12 16 24 48"
## [1] "basket: 49 | frequency: 3 | items: 1 7 49"
## [1] "basket: 50 | frequency: 6 | items: 1 2 5 10 25 50"
## [1] "basket: 51 | frequency: 4 | items: 1 3 17 51"
## [1] "basket: 52 | frequency: 6 | items: 1 2 4 13 26 52"
## [1] "basket: 53 | frequency: 2 | items: 1 53"
## [1] "basket: 54 | frequency: 8 | items: 1 2 3 6 9 18 27 54"
## [1] "basket: 55 | frequency: 4 | items: 1 5 11 55"
## [1] "basket: 56 | frequency: 8 | items: 1 2 4 7 8 14 28 56"
## [1] "basket: 57 | frequency: 4 | items: 1 3 19 57"
## [1] "basket: 58 | frequency: 4 | items: 1 2 29 58"
## [1] "basket: 59 | frequency: 2 | items: 1 59"
## [1] "basket: 60 | frequency: 12 | items: 1 2 3 4 5 6 10 12 15 20 30 60"
## [1] "basket: 61 | frequency: 2 | items: 1 61"
## [1] "basket: 62 | frequency: 4 | items: 1 2 31 62"
## [1] "basket: 63 | frequency: 6 | items: 1 3 7 9 21 63"
## [1] "basket: 64 | frequency: 7 | items: 1 2 4 8 16 32 64"
## [1] "basket: 65 | frequency: 4 | items: 1 5 13 65"
## [1] "basket: 66 | frequency: 8 | items: 1 2 3 6 11 22 33 66"
## [1] "basket: 67 | frequency: 2 | items: 1 67"
## [1] "basket: 68 | frequency: 6 | items: 1 2 4 17 34 68"
## [1] "basket: 69 | frequency: 4 | items: 1 3 23 69"
## [1] "basket: 70 | frequency: 8 | items: 1 2 5 7 10 14 35 70"
## [1] "basket: 71 | frequency: 2 | items: 1 71"
## [1] "basket: 72 | frequency: 12 | items: 1 2 3 4 6 8 9 12 18 24 36 72"
## [1] "basket: 73 | frequency: 2 | items: 1 73"
## [1] "basket: 74 | frequency: 4 | items: 1 2 37 74"
## [1] "basket: 75 | frequency: 6 | items: 1 3 5 15 25 75"
## [1] "basket: 76 | frequency: 6 | items: 1 2 4 19 38 76"
## [1] "basket: 77 | frequency: 4 | items: 1 7 11 77"
## [1] "basket: 78 | frequency: 8 | items: 1 2 3 6 13 26 39 78"
## [1] "basket: 79 | frequency: 2 | items: 1 79"
## [1] "basket: 80 | frequency: 10 | items: 1 2 4 5 8 10 16 20 40 80"
## [1] "basket: 81 | frequency: 5 | items: 1 3 9 27 81"
## [1] "basket: 82 | frequency: 4 | items: 1 2 41 82"
## [1] "basket: 83 | frequency: 2 | items: 1 83"
## [1] "basket: 84 | frequency: 12 | items: 1 2 3 4 6 7 12 14 21 28 42 84"
## [1] "basket: 85 | frequency: 4 | items: 1 5 17 85"
## [1] "basket: 86 | frequency: 4 | items: 1 2 43 86"
## [1] "basket: 87 | frequency: 4 | items: 1 3 29 87"
```

```
## [1] "basket: 88 | frequency: 8 | items: 1 2 4 8 11 22 44 88"
## [1] "basket: 89 | frequency: 2 | items: 1 89"
## [1] "basket: 90 | frequency: 12 | items: 1 2 3 5 6 9 10 15 18 30 45 90"
## [1] "basket: 91 | frequency: 4 | items: 1 7 13 91"
## [1] "basket: 92 | frequency: 6 | items: 1 2 4 23 46 92"
## [1] "basket: 93 | frequency: 4 | items: 1 3 31 93"
## [1] "basket: 94 | frequency: 4 | items: 1 2 47 94"
## [1] "basket: 95 | frequency: 4 | items: 1 5 19 95"
## [1] "basket: 96 | frequency: 12 | items: 1 2 3 4 6 8 12 16 24 32 48 96"
## [1] "basket: 97 | frequency: 2 | items: 1 97"
## [1] "basket: 98 | frequency: 6 | items: 1 2 7 14 49 98"
## [1] "basket: 99 | frequency: 6 | items: 1 3 9 11 33 99"
## [1] "basket: 100 | frequency: 9 | items: 1 2 4 5 10 20 25 50 100"
```

```
print(total_itemcount)
```

```
## [1] 482
```

Exercise 6.1.5 : For the data of Exercise 6.1.1, what is the confidence of the following association rules?

(a) {5,7} -> 2.

5 and 7 appear in two baskets together (35, 70). Of these, 2 appears in just one basket (70). Therefore, the confidence is $1/2 = 50\%$.

(b) {2,3,4} -> 5.

```
for (i in 1:100) {
  if ((i %% 2 == 0) & (i %% 3 == 0) & (i %% 4 == 0)) {
    print(i)
  }
}
```

```
## [1] 12
## [1] 24
## [1] 36
## [1] 48
## [1] 60
## [1] 72
## [1] 84
## [1] 96
```

2, 3, and 4 appear in 8 baskets together. Of these, 5 appears in just one basket (60). Therefore, the confidence is $1/8 = 12.5\%$.

Suppose the support threshold is 4. On the first pass of the PCY Algorithm we use a hash table with 11 buckets, and the set {i,j} is hashed to bucket i x j mod 11.

(a) By any method, compute the support for each item and each pair of items.

```
baskets <- list(c(1,2,3), c(2,3,4), c(3,4,5), c(4,5,6),
                c(1,3,5), c(2,4,6), c(1,3,4), c(2,4,5),
                c(3,5,6), c(1,2,4), c(2,3,5), c(3,4,6))

# Single items
for (i in 1:6) {
  count <- 0
  for (b in baskets) {
    if (i %in% b) {
      count <- count + 1
    }
  }
  print(paste("item:", i, "| baskets/support:", count))
}

# Pairs of items
p <- expand.grid(1:6, 1:6)
p2 <- p[p[,2] > p[,1],]
p2 <- p2[order(p2[,1]),]

fcount <- 0
for (i in 1:nrow(p2)) {
  dcount <- 0
  for (j in 1:length(baskets)) {
    if ((p2[i,1] %in% baskets[[j]]) & (p2[i,2] %in% baskets[[j]])) {
      dcount <- dcount + 1
    }
  }

  pair <- paste("(", p2[i,1], ",", p2[i,2], ")", sep="")
  print(paste("pair:",pair, "| baskets/support:",dcount))
}
```

```
## [1] "item: 1 | baskets/support: 4"
## [1] "item: 2 | baskets/support: 6"
## [1] "item: 3 | baskets/support: 8"
## [1] "item: 4 | baskets/support: 8"
## [1] "item: 5 | baskets/support: 6"
## [1] "item: 6 | baskets/support: 4"
## [1] "pair: (1,2) | baskets/support: 2"
## [1] "pair: (1,3) | baskets/support: 3"
## [1] "pair: (1,4) | baskets/support: 2"
## [1] "pair: (1,5) | baskets/support: 1"
## [1] "pair: (1,6) | baskets/support: 0"
## [1] "pair: (2,3) | baskets/support: 3"
## [1] "pair: (2,4) | baskets/support: 4"
## [1] "pair: (2,5) | baskets/support: 2"
## [1] "pair: (2,6) | baskets/support: 1"
## [1] "pair: (3,4) | baskets/support: 4"
```

```
## [1] "pair: (3,5) | baskets/support: 4"
## [1] "pair: (3,6) | baskets/support: 2"
## [1] "pair: (4,5) | baskets/support: 3"
## [1] "pair: (4,6) | baskets/support: 3"
## [1] "pair: (5,6) | baskets/support: 2"
```

(b) Which pairs hash to which buckets?

We can do this by pair...

```
# By item
for (i in 1:nrow(p2)) {
  hb <- (p2[i,1] * p2[i,2]) %% 11
  pair <- paste("(", p2[i,1], ",", p2[i,2], ")", sep="")
  print(paste("pair:",pair, "| bucket:",hb))
}
```

```
## [1] "pair: (1,2) | bucket: 2"
## [1] "pair: (1,3) | bucket: 3"
## [1] "pair: (1,4) | bucket: 4"
## [1] "pair: (1,5) | bucket: 5"
## [1] "pair: (1,6) | bucket: 6"
## [1] "pair: (2,3) | bucket: 6"
## [1] "pair: (2,4) | bucket: 8"
## [1] "pair: (2,5) | bucket: 10"
## [1] "pair: (2,6) | bucket: 1"
## [1] "pair: (3,4) | bucket: 1"
## [1] "pair: (3,5) | bucket: 4"
## [1] "pair: (3,6) | bucket: 7"
## [1] "pair: (4,5) | bucket: 9"
## [1] "pair: (4,6) | bucket: 2"
## [1] "pair: (5,6) | bucket: 8"
```

Or we can see which pairs are in each bucket.

```
# By bucket
for (i in 0:10) {
  hbcount <- 0
  pairs <- character()
  for (j in 1:nrow(p2)) {
    hb <- (p2[j,1] * p2[j,2]) %% 11
    if (hb == i) {
      hbcount <- hbcount + 1
      pair <- paste("(", p2[j,1], ",", p2[j,2], ")", sep="")
      pairs <- c(pairs, pair)
    }
  }

  pairs <- paste(pairs, collapse=", ")
  print(paste("bucket:", i, "| count:", hbcount, "| pair(s):", pairs))
}
```

```
## [1] "bucket: 0 | count: 0 | pair(s): "
## [1] "bucket: 1 | count: 2 | pair(s): (2,6), (3,4)"
## [1] "bucket: 2 | count: 2 | pair(s): (1,2), (4,6)"
## [1] "bucket: 3 | count: 1 | pair(s): (1,3)"
## [1] "bucket: 4 | count: 2 | pair(s): (1,4), (3,5)"
## [1] "bucket: 5 | count: 1 | pair(s): (1,5)"
## [1] "bucket: 6 | count: 2 | pair(s): (1,6), (2,3)"
## [1] "bucket: 7 | count: 1 | pair(s): (3,6)"
## [1] "bucket: 8 | count: 2 | pair(s): (2,4), (5,6)"
## [1] "bucket: 9 | count: 1 | pair(s): (4,5)"
## [1] "bucket: 10 | count: 1 | pair(s): (2,5)"
```

(c) Which buckets are frequent?

None are frequent for support threshold of 4. However, buckets 1, 2, 4, 6, and 8 are frequent for s = 2.

Based on these results, I'm thinking maybe every basket-pair combo needs to be hashed, even if it is not a unique pair overall (i.e. baskets 1 and 2 each have items 2 and 3, which gets hashed twice).

```r
# Re-hash for every pair, even if not unique
pairs_candidate <- list()
for (i in 0:10) {
  hbcount <- 0
  pairs <- character()

  for (b in baskets) {
    bcombo <- expand.grid(b, b)
    bcombo <- bcombo[bcombo[,2] > bcombo[,1],]
    bcombo <- bcombo[order(bcombo[,1]),]

    for (j in 1:nrow(bcombo)) {
      hb <- (bcombo[j,1] * bcombo[j,2]) %% 11
      if (hb == i) {
        hbcount <- hbcount + 1
        pair <- paste("(", bcombo[j,1], ",", bcombo[j,2], ")", sep="")
        pairs <- c(pairs, pair)
      }
    }
  }

  pairs <- paste(pairs, collapse=", ")
  print(paste("bucket:", i, "| count:", hbcount, "| pair(s):", pairs))
  if (hbcount >= 4) {
    pairs_candidate[length(pairs_candidate)+1] <- pairs
  }
}
```

```
## [1] "bucket: 0 | count: 0 | pair(s): "
## [1] "bucket: 1 | count: 5 | pair(s): (3,4), (3,4), (2,6), (3,4), (3,4)"
## [1] "bucket: 2 | count: 5 | pair(s): (1,2), (4,6), (4,6), (1,2), (4,6)"
## [1] "bucket: 3 | count: 3 | pair(s): (1,3), (1,3), (1,3)"
## [1] "bucket: 4 | count: 6 | pair(s): (3,5), (3,5), (1,4), (3,5), (1,4), (3,5)"
```

```
## [1] "bucket: 5 | count: 1 | pair(s): (1,5)"
## [1] "bucket: 6 | count: 3 | pair(s): (2,3), (2,3), (2,3)"
## [1] "bucket: 7 | count: 2 | pair(s): (3,6), (3,6)"
## [1] "bucket: 8 | count: 6 | pair(s): (2,4), (5,6), (2,4), (2,4), (5,6), (2,4)"
## [1] "bucket: 9 | count: 3 | pair(s): (4,5), (4,5), (4,5)"
## [1] "bucket: 10 | count: 2 | pair(s): (2,5), (2,5)"
```

Based on this new hash implementation, buckets 1, 2, 4, and 8 have support threshold of 4.

(d) Which pairs are counted on the second pass of the PCY Algorithm?

Only pairs in buckets 1, 2, 4, and 8 will be considered as candidates. Since all individual items are frequent, all pairs in these buckets will be counted on the second pass. Looking at the output above, those would be pairs (3,4), (2,6), (1,2), (4,6), (3,5), (1,4), (2,4), and (5,6).

Exercise 6.3.2: Suppose we run the Multistage Algorithm on the data of Exercise 6.3.1, with the same support threshold of 4. The first pass is the same as in that exercise, and for the second pass, we hash pairs to nine buckets, using the hash function that hashes {i, j} to bucket i + j mod 9. Determine the counts of the buckets on the second pass. Does the second pass reduce the set of candidate pairs? Note that all items are frequent, so the only reason a pair would not be hashed on the second pass is if it hashed to an infrequent bucket on the first pass.

Again, I use non-unique pairs otherwise there would be no buckets with support 4. This equates to re-hashing the 22 non-unique pairs hashed to frequent buckets 1, 2, 4, and 8 of the first pass.

```
pc <- matrix(c(3,3,2,3,3,1,4,4,1,4,3,3,1,3,1,3,2,5,2,2,5,2,
               4,4,6,4,4,2,6,6,2,6,5,5,4,5,4,5,4,6,4,4,6,4),
             nrow=22, ncol=2)

for (i in 0:9) {
  hbcount2 <- 0
  pairs2 <- character()

  for (j in 1:nrow(pc)) {
    hb <- (pc[j,1] + pc[j,2]) %% 9
    if (hb == i) {
      hbcount2 <- hbcount2 + 1
      pair2 <- paste("(", pc[j,1], ",", pc[j,2], ")", sep="")
      pairs2 <- c(pairs2, pair2)
    }
  }

  pairs2 <- paste(pairs2, collapse=", ")
  print(paste("bucket:", i, "| count:", hbcount2, "| pair(s):", pairs2))
}
```

```
## [1] "bucket: 0 | count: 0 | pair(s): "
## [1] "bucket: 1 | count: 3 | pair(s): (4,6), (4,6), (4,6)"
```

```
## [1] "bucket: 2 | count: 2 | pair(s): (5,6), (5,6)"
## [1] "bucket: 3 | count: 2 | pair(s): (1,2), (1,2)"
## [1] "bucket: 4 | count: 0 | pair(s): "
## [1] "bucket: 5 | count: 2 | pair(s): (1,4), (1,4)"
## [1] "bucket: 6 | count: 4 | pair(s): (2,4), (2,4), (2,4), (2,4)"
## [1] "bucket: 7 | count: 4 | pair(s): (3,4), (3,4), (3,4), (3,4)"
## [1] "bucket: 8 | count: 5 | pair(s): (2,6), (3,5), (3,5), (3,5), (3,5)"
## [1] "bucket: 9 | count: 0 | pair(s): "
```

Now only buckets 6, 7, and 8 are frequent on the second pass. Since all pairs in these buckets are already in frequent buckets on the first pass, they are all candidate pairs and hashed on the second pass. They are (2,4), (3,4), (2,6), and (3,5). The second pass reduced the candidate set from 8 to 4.