Reference: Big C++.

Exercises 10:

Exercise R10.1.

Recursion: Making a function calls itself again and again with simplifying the input to it.

Iteration: Using Loops to solve the problem.

Infinite Recursion: A function calls itself infinitely and never stop because it’s missing a terminate case.

Mutual Recursion: a set of cooperating functions calls each other repeatedly.

Exercise R10.2.

*n*! *n* 1! *n*

1! = (0)! \* 1 = 1 \* 1 = 1

2! = (1)! \* 2 = 1 \* 2 = 2

Permutation of a single object is itself, for instance the letter ‘a’, the permutation of it is also ‘a’

Permutation of two objects are themselves, for instance the string ‘ab’, the permutation of them are “ab” and “ba”

Exercise R10.3.

Algorithm to find the smallest value in an array:

Compare the first element and the last, remove the bigger one, do it again for the remaining array until the size of the array is one, then return that value.

int find\_smallest(int a[], int size, int start\_index, int end\_index)

{

if (size == 1)

return (a[start\_index]);

if (a[size - 1] < a[start\_index])

{

start\_index++;

return find\_smallest(a, size - 1, start\_index, end\_index);

}

else

{

end\_index--;

return find\_smallest(a, size - 1, start\_index, end\_index);

}

}

Exercise R10.4.

Find the smallest element in the array. Swap the first element with this value in the array. Repeat until the whole array is sorted.

Exercise R10.5.

If the subset == 1, return 1. Else, return generate\_subset (n – 1).

Exercise R10.6.