Problem-90

Problem Statement

(a) Multiply $(1+x)\left(1+x^2\right)$.

(b) Multiply $(1+x)\left(1+x^2\right)\left(1+x^4\right)\left(1+x^8\right)$.

(c) Compute $\left(1+x+x^2+x^3\right)^2$.

(d) Compute $\left(1+x+x^2+x^3+\ \dots\ +x^9+x^{10}\right)^2$.

(e) Find the coefficient of x^{29} and x^{30} in $\left(1+x+x^2+x^3+\ \dots\ +x^9+x^{10}\right)^3$.

(f) Multiply $(1-x)\left(1+x+x^2+x^3+\ \dots\ +x^9+x^{10}\right)$.

(g) Multiply $(a+b)\left(a^2-ab+b^2\right)$.

(h) Multiply $\left(1-x+x^2-x^3+x^4-x^5+x^6-x^7+x^8-x^9+x^{10}\right)$ by $\left(1+x+x^2+x^3+x^4+x^5+x^6+x^7+x^8+x^9+x^{10}\right)$.

Solution to Part-a

$$\begin{array}{r} x+1 \\ \times \quad x^2+1 \\ \hline x+1 \\ x^3+x^2 \quad\quad \\ \hline x^3+x^2+x+1 \end{array}$$

Solution to Part-b

From part-a we have $(1+x)\left(1+x^2\right)=1+x+x^2+x^3$.

$$\begin{array}{r} x^3+x^2+x+1 \\ \times \quad\quad\quad\quad x^4+1 \\ \hline x^3+x^2+x+1 \\ x^7+x^6+x^5+x^4 \quad\quad\quad\quad\quad\quad \\ \hline x^7+x^6+x^5+x^4+x^3+x^2+x+1 \end{array}$$

Similar to the above, $\left(x^7+x^6+x^5+x^4+x^3+x^2+x+1\right)\left(1+x^8\right)=\sum_{n=0}^{15} x^n$.

Solution to Part-c

$$\begin{array}{r} x^3+x^2+x+1 \\ \times \quad x^3+x^2+x+1 \\ \hline x^3+x^2+x+1 \\ x^4+x^3+x^2+x \quad\; \\ x^5+x^4+x^3+x^2 \qquad\quad\; \\ x^6+x^5+x^4+x^3 \qquad\qquad\qquad \\ \hline x^6+2x^5+3x^4+4x^3+3x^2+2x+1 \end{array}$$

In regards to how many times they occur, we note that left and right of x^3 are mirror reflections. For example, there are as many x^5 as there are x. We may write the product as

$$\sum_{n=0}^{2}(n+1)\cdot x^n \;+\; 4\cdot x^3 \;+\; \sum_{n=4}^{6}(6-n+1)\cdot x^n$$

Solution to Part-d

From the observation in part-c, the product here should be

$$\sum_{n=0}^{9}(n+1)\cdot x^n \;+\; 11\cdot x^{10} \;+\; \sum_{n=11}^{20}(20-n+1)\cdot x^n$$

In other words the product is

$$x^{20}+2x^{19}+3x^{18}+\;\ldots\;+10x^{11}+11x^{10}+10x^9+9x^8+\;\ldots\;+3x^2+2x+1$$

Solution to Part-e

We observe that the coefficient of x^{29} in $\left(1+x+x^2+x^3+\;\ldots\;+x^9+x^{10}\right)^3$ is the count of ways we can get 29 from adding three integers between 0 and 10 inclusive. In other words, it is the number of tuples (a,b,c) where $0 \le a,b,c \le 10$ and $a+b+c=29$. There are three such tuples: $(9,10,10)$, $(10,9,10)$, and $(10,10,9)$. So, the coefficient of x^{29} is 3. There is only one tuple whose elements sum to 30, namely the tuple $(10,10,10)$; so the coefficient of x^{30} is 1.

Java code that follows, gives the below output for $\left(1+x+x^2+x^3+\;\ldots\;+x^9+x^{10}\right)^3$ and we see that the coefficients of x^{29} and x^{30} are indeed 3 and 1 respectively.

```
1  + 3 * x^1  + 6 * x^2  + 10 * x^3  + 15 * x^4  + 21 * x^5  + 28 * x^6
+ 36 * x^7  + 45 * x^8  + 55 * x^9  + 66 * x^10  + 75 * x^11  + 82 * x^12
+ 87 * x^13  + 90 * x^14  + 91 * x^15  + 90 * x^16  + 87 * x^17
+ 82 * x^18  + 75 * x^19  + 66 * x^20  + 55 * x^21  + 45 * x^22
+ 36 * x^23  + 28 * x^24  + 21 * x^25  + 15 * x^26  + 10 * x^27
+ 6 * x^28  + 3 * x^29  + 1 * x^30
```

```
  public static void main(String[] args) {
    int[] polynomial1 = new int[11];
    Arrays.fill(polynomial1, 1);
    int[] polynomial2 = new int[11];
    Arrays.fill(polynomial2, 1);
    // we get the square of 1+x+x^2+...+x^10 here
    int[] square = multiply( polynomial1, polynomial2 );
    int[] cube = multiply( square, polynomial1 );
    printPolynomial(cube);
  }

  private static int[] multiply(int[] polynomial1, int[] polynomial2) {
    int m = polynomial1.length;
    int n = polynomial2.length;
    int[] product = new int[m+n];
    for (int i = 0; i < m; ++i) {
      for (int j = 0; j < n; ++j) {
        product[i+j] += polynomial1[i]*polynomial2[j];
      }
    }
    // get rid of extra zeros for the higher powers
    int k = m+n-1;
    while (k > 0 && product[k] == 0) {
      --k;
    }
    int[] result = new int[k+1];
    for (int i = 0; i <= k; ++i) {
      result[i] = product[i];
    }
    return result;
  }
```

```
private static void printPolynomial(int[] poly) {
  System.out.println();
  System.out.print( String.format( "%d", poly[0] ) );
  for (int i = 1; i < poly.length; ++i) {
    System.out.print( String.format( "  + %d * x^%d", poly[i], i ) );
  }
  System.out.println();
}
```