Keith Arithmetic - Alan Yan

We will find a closed form of $S(f,\ k)$.

Consider the sequence of polynomials $P_0,\ P_1,\ P_2,\ \ldots$ defined by

$$P_0(x) = f(x), \qquad P_{i+1}(x) = P_i(x) - P_i(x+2^i)$$

Lemma 1. $P_n(x) = \sum_{i=0}^{2^n-1} (-1)^{s(n)} f(x+i)$

Proof. We induct on n.

Base Case: $n = 0$, $P_0(x) = f(x) = \sum_{i=0}^{0} f(x+i)$.

Inductive Step: Suppose the formula is true for n. Observe that

$$P_{n+1}(x) = P_n(x) - P_n(x+2^n) = \sum_{i=0}^{2^n-1} (-1)^{s(n)} f(x+i) - \sum_{i=0}^{2^n-1} (-1)^{s(n)} f(x+i+2^n)$$

$$= \sum_{i=0}^{2^n-1} (-1)^{s(n)} f(x+i) + \sum_{i=2^n}^{2^{n+1}-1} (-1)^{s(n)} f(x+i) = \sum_{i=0}^{2^{n+1}-1} (-1)^{s(n)} f(x+i)$$

completing the induction.

Lemma 2. $deg\ P_n(x) = max(\deg f - n,\ 0)$

Proof. Follows from the fact that the operation $f \to f(x+c) - f(x)$ decreases the degree by 1.

Observe that $S(f,\ k) = P_k(0)$.

Lemma 3. If $f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots$. Then the leading coefficient of $f(x) - f(x+c)$ is $-a_n nc$.

Proof. Only the x^n will contribute to the leading coefficient. The leading coefficient of $a_n x^n - a_n(x+c)^n$ is $-a_n nc$.

Let $d = deg\, f$. Now, consider two cases:

Case 1: $k > d$. From Lemma 2, $P_k(x) = 0$. So our answer will be 0.

Case 2: $k = d$. Then $P_k(x)$ is constant. From repeated application of Lemma 3, we have that

$$P_k(x) = (-1)^d d! 2^{\binom{d}{2}}$$

We can now solve the problem.

Algorithm:

1. $ret = 0$.
2. For $1 \le i \le N$, iterate the rest of the steps
3. If $deg\, p_i < k_i$, add 1 to ret.
4. If $deg\, p_i = k_i$, add $\binom{d}{2} + v(d!) + v(c)$ to ret.

The flag will be **ret**. My code is on the next page.

```python
def L(n):
      ret =0
      while(n % 2 == 0):
            ret += 1
            n /= 2
      return ret

def C(n):
      return n * (n-1) / 2

def s(n):
      ret = 0
      while(n > 0):
            if(n % 2 == 1):
                        ret += 1
            n //= 2
      return ret
def v(n):
      return n - s(n)

def S(coef, deg, k):
      if(k > deg):
            return 1
      if(coef == 0):
            return 1
      return C(deg) + v(deg) + L(coef)

r = open("Keith", "r")
data = r.read()

ret = 0
lines = data.splitlines()
for i in range(2000):
      k = int(lines[2*i])
      seq = lines[2*i+1]
      arr = seq.split(" ")
      coef = int(arr[0])
      deg = len(arr) - 1
      ret += S(coef, deg, k)
print(ret)
```