

چندجمله‌ای چندمتغیره بولی

مقدمات

در این سؤال به پیاده سازی چندجمله‌ای‌های چندمتغیره بولی می پردازیم که یکی از ابزارهای پرکاربرد در حوزه کدینگ، رمزنگاری و تئوری محاسبه می باشد. چندجمله‌ای چندمتغیره بولی، توابعی هستند که به شکل

$$\{0, 1\}^n \rightarrow F$$

تعریف می شوند به طوری که نقاط را از یک فضای بولی چندمتغیره به یک عدد در مجموعه خروجی F (در این سؤال فرض کنید F مجموعه ی اعداد صحیح می باشد) نسبت می دهند و در صورت ثابت نگه داشتن n -
1 نقطه، و مقداری به آنها، عبارت باقی مانده تشکیل یک چندجمله‌ای یک متغیره می دهد. به عنوان مثال به چندجمله‌ای‌های چندمتغیره بولی زیر توجه کنید:

$$p_1(x_1, x_2, x_3) = 13x_1x_2 + x_3^{10} - 8$$

که در اینجا داریم:

$$p_1(0, 1, 1) = -7$$

$$p_1(1, 1, 0) = 5$$

یا چندجمله‌ای

$$p_2(x_1, x_2, x_3, x_4, x_5) = 4x_3^2x_5 - 3x_1x_2^4x_5$$

که در اینجا داریم:

$$p_2(1, 0, 1, 0, 1) = 4$$

$$p_2(1, 1, 1, 1, 1) = 1$$

نکته: بدیهی است که توان رسانی در متغیرهای بولی تغییری در مقادیر خروجی ایجاد نمی‌کند، منتها توان‌ها در برخی سناریوها کاربردهای خاص خودشان را دارند، لذا باید پیاده سازی شوند.

در ادامه به چند تعریف توجه بفرمایید:

درجه نسبت به یک متغیر: درجه‌ی یک چندجمله‌ای چندمتغیره نسبت به متغیر x_i ، برابر است با بیشینه‌ی درجه‌ی متغیر x_i در عبارات چندجمله‌ای.

درجه‌ی کل: درجه‌ی کل چندجمله‌ای برابر است با بیشینه‌ی درجات عبارات مختلف یک چندجمله‌ای

به عنوان مثال در چندجمله‌ای

$$p_3(x_1, x_2, x_3) = x_2 x_3^4 + 6x_1^2 + 3x_1^3 x_2 x_3^9 - x_1 x_2 x_3$$

درجه‌ی متغیر x_1 برابر است با 3، درجه‌ی متغیر x_2 برابر است با 1 و درجه‌ی متغیر x_3 برابر است با 9. درجه‌ی عبارت

$$x_2 x_3^4$$

برابر است با 5، درجه عبارت

$$6x_1^2$$

برابر است با 2، درجه عبارت

$$3x_1^3 x_2 x_3^9$$

برابر است با 13 و درجه عبارت

$$-x_1 x_2 x_3$$

برابر است با 3. همچنین درجه‌ی کل چندجمله‌ای برابر است با

$$\max\{5, 2, 13, 3\} = 13$$

همچنین تعریف می‌کنیم *جمع کل چندجمله‌ای* برابر است با

$$\sum_{(x_1, \dots, x_n) \in \{0,1\}^n} p(x_1, \dots, x_n)$$

پیاده سازی

با توجه به تعاریف بالا، تابع main زیر به شما تحویل داده شده است:

Copy Java

```

1
2 public static void main(String[] args) {
3     Scanner sc= new Scanner(System.in);
4     MultivariatePolynomial poly = new MultivariatePolynomial(sc);
5     Point point = new Point(sc);
6     int variable_count = poly.get_variable_count();
7     System.out.println(poly.calculate(point));
8     for (int i = 1; i < variable_count+1; i++) {
9         System.out.println(poly.get_degree(i));
10    }
11    System.out.println(poly.get_degree());
12    System.out.println(poly.get_sum());
13 }
```

کلاس ها، داده ساختارها و توابع لازم را برای کار کردن قطعه کد بالا پیاده سازی کنید.

در ادامه به شکل ورودی و خروجی کد بالا دقت کنید:

ورودی

در این بخش قالب ورودی و محدودیت‌های آن توضیح داده شود.

در ورودی به ترتیب ۱- یک چندجمله‌ای چند متغیره ی بولی ۲- یک نقطه (در فضای چندبعدی ورودی تابع)

داده خواهد شد. نحوه‌ی گرفتن چند جمله‌ای بدین شکل است که عبارات آن به ترتیب زیر هم نوشته خواهند شد. در هر عبارت، توان هر متغیر نوشته شده است. همچنین عدد اول هر عبارت (متغیر صفرم) را

ضریب آن عبارت در نظر بگیرید. پس از دریافت چندجمله‌ای، یک خط خالی و سپس یک نقطه در فضای n بعدی ورودی دریافت می‌شود، این نقطه به صورت n عدد که نمایانگر n مولفه ی آن است نمایش داده می‌شود.

خروجی

در این بخش قالب خروجی کد کاربران توضیح داده شود.

فرمت خروجی کد به همان صورتی است که در کد تابع main آورده شده است. در صورت این که بعد نقطه ورودی و ورودی تابع یکسان نباشد invalid input چاپ شده و اجرای برنامه متوقف می‌شود.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده می‌شود.

ورودی نمونه ۱

```
1 0 5 0 2 4
5 1 2 0 0 0
-2 0 0 2 2 2
9 0 0 0 0 0

1 0 1 1 0
```

Copy

که این ورودی معادلا برابر است با چندجمله‌ای

$$p(x_1, x_2, x_3, x_4, x_5) = x_2^5 x_4^2 x_5^4 + 5x_1 x_2^2 - 2x_3^2 x_4^2 x_5^2 + 9$$

و نقطه ی داده شده معادل است با نقطه‌ی

$$(x_1, x_2, x_3, x_4, x_5) = (1, 0, 1, 1, 0)$$

خروجی نمونه ۱

```
9
1
5
2
2
4
11
324
```

[Copy](#)

ورودی نمونه ۲

```
1 0 5 0 2 4
5 1 2 0 0 0
-2 0 0 2 2 2
9 0 0 0 0 0

1 0 1
```

[Copy](#)

خروجی نمونه ۲

```
invalid input
```

[Copy](#)

تضمین می‌شود که تنها ورودی نامناسب داده شده، ورودی است که ابعاد نقطه‌ی داده شده با ابعاد چندجمله‌ای مطابق نباشد.

چندجمله‌ای چندمتغیره بولی جنریک

#مقدمات

در سؤال قبل با چندجمله‌ای‌های چندمتغیره بولی آشنا شدیم. همانطور که دیدید ضرایب چندجمله‌ای و برد این توابع همگی از جنس اعداد صحیح بودند. در این بخش می‌خواهیم این شرط را برداریم و کلاس را به شکل generic پیاده‌سازی کنیم. به عنوان مثال کلاس ما باید بتواند چندجمله‌های چندمتغیره از جنس اعداد گویا بسازد، به طوری که دامنه آن باینری و ضرایب و برد آن از جنس اعداد گویا باشد (محدودیتی روی توان‌های چندجمله‌ای گذاشته نمی‌شود و همچنان اعداد صحیح نامنفی می‌باشند). برای این سؤال تابع Main زیر در نظر گرفته شده است:

Copy Java

```

1 public static void main(String[] args) {
2     Scanner sc = new Scanner(System.in);
3     FF.setP(Integer.parseInt(sc.nextLine()));
4     MultivariatePolynomial<FF> poly = new MultivariatePolynomial<>(sc);
5     Point point = new Point(sc);
6     int variable_count = poly.get_variable_count();
7     System.out.println(poly.calculate(point));
8     for (int i = 1; i < variable_count + 1; i++) {
9         System.out.println(poly.get_degree(i));
10    }
11    System.out.println(poly.get_degree());
12    System.out.println(poly.get_sum());
13 }
```

برای پیاده‌سازی کلاس MultivariatePolynomial عیناً از قطعه کد زیر استفاده کنید و فقط درون کلاس MultivariatePolynomial را تغییر دهید:

Copy Java

```

1 interface MyField<T> {
2     static Object getInstance(int value) {
3         return new FF(value);
4     }
5     T multiply(T b);
6     T add(T b);
7 }
```

```

8
9 class MultivariatePolynomial<FF extends MyField<FF>> {
10 ...
11 }

```

که کلاس FF کلاس میدان محدود است که در ادامه درباره‌ی آن توضیح می‌دهیم. کلاس میدان محدود بسیار شبیه به کلاس اعداد صحیح است، با این تفاوت که مجموعه‌ی این اعداد محدود است. همانطور که می‌دانیم اعداد صحیح به صورت زیر تعریف می‌شوند:

$$Z = \{..., -3, -2, -1, 0, 1, 2, 3, ...\}$$

که اعمال جمع و ضرب روی آن به صورت استاندارد تعریف شده است. اما یک میدان متناهی مجموعه‌ای است که به صورت

$$F = \{0, 1, 2, 3, ..., p - 1\}$$

تعریف می‌شود که p در آن یک عدد اول است. تمام عملیات های ضرب و جمع و تفریق به پیمانه‌ی p انجام می‌شود. (عملیات تقسیم به شیوه‌ای متفاوت انجام می‌شود که در این سؤال به آن نیازی نداریم) به عنوان مثال برای $p=97$ داریم:

$$190 + 32 = 28$$

یا

$$3 \times 87 = 67$$

لذا در این مجموعه اعداد منفی یا اعداد بیشتر از p معنایی ندارند و در عمل باید به اعداد مجموعه FF نگاشت بشوند.

پیاده سازی

در این بخش، ما یک کلاس MultivariatePolynomial<FF> را پیاده‌سازی می‌کنیم که از کلاس MyField<FF> ارث‌برداری می‌کند.

در این سوال مسد سوال قبل باید با توجه به ضعه حد راهنمای داده سده حلسهایی را پیاده حید ده بتوان با آن تابع Main داده شده را اجرا کرد. تاکید مجدد: ضرایب چندجمله‌ای و بُرد آن میدان محدودی است که ورودی p آن را تعیین می‌کند. تضمین می‌شود که این شرط برای تمامی ضرایب ورودی رعایت می‌شود. دقت کنید: جمع کل تابع نیز در میدان FF محاسبه می‌گردد.

ورودی و خروجی

ورودی‌ها و خروجی‌ها درست مانند سوال قبل می‌باشند و در صورت این که بعد نقطه ورودی و ورودی تابع یکسان نباشد `invalid input` چاپ شده و اجرای برنامه متوقف می‌شود. عدد اول p در خط اول برنامه داده خواهد شد.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده می‌شود.

ورودی نمونه ۱

97

1 0 5 0 2 4

5 1 2 0 0 0

9 0 0 0 0 0

1 0 1 1 0

Copy

که این ورودی معادلا برابر است با چندجمله‌ای

$$p(x_1, x_2, x_3, x_4, x_5) = x_2^5 x_4^2 x_5^4 + 5x_1 x_2^2 + 9$$

و نقطه‌ی داده شده معادل است با نقطه‌ی

$$(x_1, x_2, x_3, x_4, x_5) = (1, 0, 1, 1, 0)$$

و $p=97$ می‌باشد.

خروجی نمونه ۱

خروجی نمونه ۱

Copy

9
1
5
0
2
4
11
41

ورودی نمونه ۲

Copy

31
6 0 0 0 4 1 7 6
11 1 2 0 0 0 3 1
15 1 5 1 0 0 1 0

1 1 1 0 0 1 0

خروجی نمونه ۲

Copy

15
1
5
1
4
1
7
6
18
8

تضمین می شود که تنها خطای ممکن در گرفتن ورودی، عدم تطابق ابعاد نقطه‌ی ورودی و چندجمله‌ای می‌باشد.

**** توجه: **** تست کیس ، شماره ۲۰ ، امتیاز ۰ می‌باشد.

سوال 1 از 10 سوال

کمی تشریحی

۱. نوع داده شمارشی (enumeration) مدرک تحصیلی (Degree) شامل سه مقدار MS, BS و PHD را در نظر بگیرید. با فرض اینکه جاوا امکان تعریف این نوع را با کلیدواژه enum نداشته باشد، نوع داده ی مذکور را با سایر امکانات جاوا به گونهای پیاده سازی نمایید که دقیقا همان خواص enum را داشته باشد. (۴۰ نمره)

۲. در قطعه کد زیر خروجی "mystery(abcdefg)" چیست؟ (۲۰ نمره)

```

1  private String mystery(String s) {
2      String result = "";
3      int len = s.length();
4      int j = 0;
5      int k = 9;
6      while (j < k) {
7          if (j < 4) {
8              result += s.charAt(k % len);
9          }
10         if (j / 2 != 1) {
11             result += s.charAt(j % len);
12         }
13         j++;
14         k--;
15     }
16     return result;
17 }
```

۳. برنامه ی زیر برای محاسبه ی نزدیک ترین عدد صحیح به میانگین N عدد صحیح که از کاربر گرفته می شود طراحی شده است. پس از مطالعه ی این برنامه به پرسش های زیر پاسخ دهید:

```

1  public class Mean{
```

```
2      public static void main(String[] args){
3          int N = args[0];
4          int[] a = new int[N];
5          for(int i = 0; i < N; i++)
6              a[i] = StdIn.readInt();
7          int sum = a[0];
8          for(int i = 1; i <= N; i++)
9              sum = sum + a[i];
10         StdOut.println("Mean: " + sum/N);
11     }
12 }
```

1- یک باگ که باعث می‌شود کد compile نشود را شرح دهید. (۱۰ نمره)

2- یک باگ که باعث می‌شود کد در runtime خطا داشته باشد را شرح دهید. (۱۰ نمره)

3- یک باگ که باعث می‌شود خروجی برنامه در بسیاری از ورودی‌ها اشتباه باشد را شرح دهید. (۱۰ نمره)

۴- راهکاری برای کاهش مصرف Memory این برنامه پیشنهاد دهید. (۱۰ نمره)