Лабораторная работа № 1. Алгоритм Е

Лит-ра: Алгебра, лек. 6.

Реализация алгоритма Евклдиа для кольца $\mathbb{Q}[x]$.

В [2]:

```
1 def quo_rem_poly(f,q):
2     K=f.parent()
3     n=0
4     while f.degree()>=q.degree():
5         ni=K(f.lt()/q.lt())
6         n=n+ni
7         f=f-ni*q
8     return (n,f)
```

1. Какого типа данные подаются на вход? Какого типа данные возвращ строку алгоритма.

На вход подаются данные строкового типа - многочлены. Возвращаются да это частное и остаток от деления.

В первой строке задается функция и входные данные - два многочлена, де второй строке определяется поле, в котором мы будем работать. Далее мы выражения и оно напрямую зависит от коэффициентов в делимом. В 4-й ст условие: наш многочлен f будет меняться и алгоритм будет выполняться, п степени q. Затем отдельно прописывается результат от деления старшего м старший моном делитель, а также кольцо, которому должен будет принадл строке формируется частное, оно будет выглядеть как многочлен, состоящ старших мономов делимого и делителя. И далее записывается остаток, кот следующей операции.

2. Проверить работу для $\mathbb{Q}[x]$

В [3]:

```
f=x^9+x+2+4*x
q=x-1
quo_rem_poly(QQ[x](f),QQ[x](q))
```

Out[3]:

```
(x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 6, 8)
```

Проверка: $f = uq + r$.

В [4]:

```
(u,r)=quo_rem_poly(QQ[x](f),QQ[x](q))
QQ[x](f-u*q-r)
```

Out[4]:

```
0
```

Встроенная в Sage реализация:

В [6]:

```
QQ[x](f).quo_rem(QQ[x](q))
```

Out[6]:

```
(x^2 + x + 2, 4)
```

3. Проверить работу для $\mathbb{Q}[t]$

В [5]:

```
var("t")
f=t^3+t+2
q=t-1
quo_rem_poly(QQ[t](f),QQ[t](q))
```

Out[5]:

```
(t^2 + t + 2, 4)
```

Вопрос: зачем нужно писать var("t")?

Необходимо писать "var("t")" для того, чтобы Sage определил ранее не зада

4. Проверить работу над $GF(p)[x]$.

В [16]:

```
f=7*x^3+x+2
q=x-1
k=GF(5)
quo_rem_poly(k[x](f),k[x](q))
```

Out[16]:

```
(2*x^2 + 2*x + 3, 0)
```

Проверка: $f = uq + r$.

В [17]:

```
(u,r)=quo_rem_poly(k[x](f),k[x](q))
k[x](f)-u*k[x](q)-r
```

Out[17]:

```
0
```

Вопрос: почему нельзя писать u*q?

Нельзя писать u*q, поскольку в таком случае операция будет проведена не определяться по заданным многочленам

5. Проверить работу над $k[y]$, где k -- поле частных кольца $\mathbb{Q}[x]$.

В [18]:

```
var("x,y")
f=x^2*y^2+(x-1)*y+1
q=x*y-1
k=FractionField(QQ[x])
quo_rem_poly(k[y](f),k[y](q))
```

Out[18]:

```
(x*y + (2*x - 1)/x, (3*x - 1)/x)
```

В [19]:

```
(u,r)=quo_rem_poly(k[y](f),k[y](q))
k[y](f)-u*k[y](q)-r
```

Out[19]:

```
0
```

6. Напишите программу для тестирования алгоритма в $\mathbb{Q}[x]$.

В [23]:

```
def test_quo_rem():
    K=QQ[x]
    u=QQ[x].random_element()
    r=QQ[x].random_element()
    q=QQ[x].random_element()
    f=u*q+r
    print((f,q))
    (uu,rr)=quo_rem_poly(K(f),K(q))
    if u==uu and rr==r:
        ans=True
    else:
        ans=False
    return ans
```

В [25]:

```
test_quo_rem()
```

```
(-1/2*x^2 + 77/26*x + 2/13, x - 4)
```

Out[25]:

```
False
```

Вопрос: должен ли алгоритм работать при $\partial q = 0$ и при $\partial q = -1$? Как "дурака"?

Нет, т.к при степени делителя 0 и -1 алгоритм не будет работать. Для того, ч ошибку, нужно вставить строку с проверкой того, чтобы степени выбранного многочлена q не была равна 0 или -1.

7. Оцените время, потребное на работу реализации алгоритма, при "боль

В [21]:

```
var("x")
f=sum([(n+1)*x^n for n in range(10^3)])
q=sum([x^n for n in range(20)])
%timeit quo_rem_poly(QQ[x](f),QQ[x](q))
```

```
360 ms ± 25.7 ms per loop (mean ± std. dev. of 7 runs, 1 loop ea
```

Замечание. Elapsed real time, real time, wall-clock time, wall time, or walltime is the start of a computer program to the end. In other words, it is the difference be task finishes and the time at which the task started.

Время для расчета подобных степеней я считаю весьма малым, но конечно время работы будет увеличиваться

В [ ]:

```

```