АЛГОРИТМИКА

ЛАБОРАТОРНЫЕ РАБОТЫ

ДЛЯ СТУДЕНТОВ ПМ4

ЛЕКТОР Аль-Натор М.С.

2010 г.

ЛАБОРАТОРНАЯ РАБОТА 1

Рекурсивные и итерационные алгоритмы

1. Примеры рекурсивных алгоритмов

a) Рекурсивные числа Фибоначчи.

Числа Фибоначчи: $F_0 = 0$, $F_1 = 1$, и $F_n = F_{n-2} + F_{n-1}\ \forall\ n \geq 2$.

```
FIB(n)
1  ▷ возвращает F_n
2  if n ≤ 1
3     then return(n)
4  else return(FIB(n − 1) + FIB(n − 2))
```

b) Рекурсивный максимум.

```
MAXIMUM(n)
1  ▷ возвращает max A[1..n]
2  if n ≤ 1
3     then return(A[1])
4  else return(max(MAXIMUM(n − 1), A[n]))
```

c) Рекурсивное умножение натуральных чисел.

```
MULTIPLY(x, y)
1  ▷ MULTIPLY(x, y) возвращает xy, где x, y ∈ ℕ
2  if y = 0
3     then return(0)
4     else  if y нечетно
5              then return(MULTIPLY(2x, ⌊y/2⌋) + x)
6              else  return(MULTIPLY(2x, ⌊y/2⌋))
```

2. Примеры итерационных алгоритмов

a) Итерационные числа Фибоначчи.

```
FIB(n)
 1  ▷ возвращает F_n
 2  if n = 0
 3      then return(0)
 4      else  a ← 0
 5            b ← 1
 6            i ← 2
 7            while i ≤ n
 8                  do c ← a + b
 9                     a ← b
10                     b ← c
11                     i ← i + 1
12   return(b)
```

b) Итерационный максимум.

```
MAXIMUM(A, n)
1  ▷ возвращает max A[1..n]
2  m ← A[1]
3  i ← 2
4  while i ≤ n
5        do if A[i] > m
6              then m ← A[i]
7           i ← i + 1
8  return(m)
```

c) Итерационное умножение натуральных чисел.

```
MULTIPLY(x, y)
1  ▷ MULTIPLY(x, y) возвращает xy, где x, y ∈ ℕ
2  p ← 0
3  while y > 0
4        do if y нечетно
5              then p ← p + x
6           x ← 2x
7           y ← ⌊y/2⌋
8  return(p)
```

ЛАБОРАТОРНАЯ РАБОТА 2

Алгоритмы решения задачи о рюкзаке

1) Наивный алгоритм

SubSet-Sum1(A)

1 ▷ **INPUT**: множество натуральных чисел $\{a_1, \ldots, a_n\}$ и натуральное число s
2 ▷ **OUTPUT**: $\epsilon_i \in \{0, 1\}, 1 \leqslant i \leqslant n$, такие, что $\sum_{i=1}^{n} \epsilon_i a_i = s$
3 Для каждого вектора $(\epsilon_1, \epsilon_2, ..., \epsilon_n) \in {\mathbb{Z}_2}^n$
4 **do** $l = \sum_{i=1}^{n} \epsilon_i a_i$
5 **if** $s = l$
6 **return**$(\epsilon_1, \epsilon_2, ..., \epsilon_n)$ — решение
7 **return**(не существует решения)

Алгоритм требует $O\left(2^n\right)$ шагов, и поэтому не является эффективным.

2) Алгоритм Meet-in-the-middle

SubSet-Sum2(A)

1 ▷ **INPUT**: множество натуральных чисел $\{a_1, \ldots, a_n\}$ и натуральное число s
2 ▷ **OUTPUT**: $\epsilon_i \in \{0, 1\}, 1 \leqslant i \leqslant n$, такие, что $\sum_{i=1}^{n} \epsilon_i a_i = s$
3 Положить $t \leftarrow \lfloor n/2 \rfloor$
4 Создать таблицу $\left(\sum_{i=1}^{t} \epsilon_i a_i, (\epsilon_1, \epsilon_2, ..., \epsilon_t)\right)$ для $(\epsilon_1, \epsilon_2, ..., \epsilon_t) \in {\mathbb{Z}_2}^t$.
Отсортировать ее по первой компоненте.
5 Для каждого $(\epsilon_{t+1}, \epsilon_{t+2}, ..., \epsilon_n) \in {\mathbb{Z}_2}^{n-t}$
6 **do** Вычислить $l = s - \sum_{i=t+1}^{n} \epsilon_i a_i$,
7 проверить с помощью бинарного поиска, является ли l
первой компонентой какой-либо ячейки таблицы
8 **if** $l = \sum_{i=1}^{t} \epsilon_i a_i$
9 **return**$(\epsilon_1, \epsilon_2, ..., \epsilon_n)$ — решение
10 **return**(не существует решения)

Алгоритм требует $O\left(n2^{n/2}\right)$ шагов, и поэтому не является эффективным.

3) Эффективным алгоритм для решения проблемы сверхвозрастающего рюкзака

SubSet-Sum3(B)

1 ▷ **INPUT**: Сверхвозрастющий рюкзак $\mathcal{B} = \{b_1, \ldots, b_n\}$
и натуральное число s, которое является суммой элементов из $\mathcal{B}$
2 ▷ **OUTPUT**: $\epsilon_i \in \{0, 1\}, 1 \leqslant i \leqslant n$, такие, что $\sum_{i=1}^{n} \epsilon_i b_i = s$
3 Положить $i \leftarrow n$
4 **while** $i \geq 1$
5 **do if** $s \geq b_i$
6 **then** $\epsilon_i \leftarrow 1$
7 $s \leftarrow s - b_i$
8 **else** $\epsilon_i \leftarrow 0$
9 $i \leftarrow i - 1$
10 **return**$(\epsilon_1, \ldots, \epsilon_n)$.

ЛАБОРАТОРНАЯ РАБОТА 3

Алгоритмы в $\mathbb{Z}$

1) Алгоритм Евклида нахождения НОД двух целых чисел

GCD(a, b)

```
1  ▷ INPUT: два неотрицательных целых чисел a ≥ b ≥ 0
2  ▷ OUTPUT: наибольший общий делитель a, b
3  while b ≠ 0
4        do r ← a  mod b
5           a ← b
6           b ← r
7  return(a)
```

Сложность алгоритма $O(\lg^2 n)$ битовых операций.

2) Расширенный алгоритм Евклида (РАЕ)

EXTGCD(a, b)

```
 1  ▷ INPUT: два неотрицательных целых чисел a ≥ b ≥ 0
 2  ▷ OUTPUT: d — НОД a, b и x, y ∈ ℤ: ax + by = d
 3  if b = 0
 4     then d ← a
 5          x ← 1
 6          y ← 0
 7  return(d, x, y)
 8  while b > 0
 9        do q ← ⌊a/b⌋
10           r ← a − qb
11           x ← x2 − qx1
12           y ← y2 − qy1
13           a ← b
14           b ← r
15           x2 ← x1
16           x1 ← x
17           y2 ← y1
18           y1 ← y
19  d ← a
20  x ← x2
21  y ← y2
22
23  return(d, x, y)
```

Сложность алгоритма $O(\lg^2 n)$ битовых операций.

Квадратичные вычеты по модулю n и квадратные корни по модулю n

Рассмотрим уравнение

$$x^2 \equiv a \mod n \quad \text{в} \quad \mathbb{Z}_n^*. \tag{1}$$

Если (1) разрешимо, то a называется *квадратичным вычетом* по модулю n, а x — *квадраным корнем* по модулю n для a.

Пусть $Q_n \subset \mathbb{Z}_n^*$ — множество всех квадратичных вычетов по модулю n. Имеем

$$Q_n = (\mathbb{Z}_n^*)^2. \tag{2}$$

Положим

$$\overline{Q}_n = \mathbb{Z}_n^* \setminus Q_n. \tag{3}$$

В дальнейшем число n предполагается нечетным.

Теорема 1 о квадратичных вычетах и квадратных корнях по модулю простого нечетного числа $n = p$.

(i) $|Q_p| = \frac{p-1}{2}$. В частности,

$$|Q_p| = \left|\overline{Q}_p\right| = \frac{p-1}{2}. \tag{4}$$

(ii) Если x — квадратный корень по модулю p для a, то $-x$ также квадратный корень по модулю p для a, и любой квадратный корень y для a удовлетворяет тождеству $y \equiv \pm x \mod p$

(iii) Имеем

$$a^{\frac{p-1}{2}} \equiv \pm 1 \mod p \quad \text{для всех} \quad \mathbb{Z} \ni a \not\equiv 0 \mod p.$$

(iv) Имеем

$$a \in Q_p \Leftrightarrow a^{\frac{p-1}{2}} \equiv 1 \mod p. \tag{5}$$

Алгоритм 1 нахождения квадратных корней для a по модулю простого нечетного числа $n = p$.
INPUT: нечетное простое число p и целое число a: $1 \leq a \leq p-1$.
OUTPUT: два квадратных корня для a по модулю p, при условии, что $a \in Q_p$.
1. Вычислить символ Лежандра $\left(\frac{a}{p}\right) =: a^{\frac{p-1}{2}} \mod p$. Если $\left(\frac{a}{p}\right) = -1$, то a не имеет квадратных корней (т.е. $a \notin Q_p$) и остановиться.
2. Выбрать случайно $b \in \mathbb{Z}$: $1 \leq b \leq p-1$, пока не будет найдено b: $\left(\frac{b}{p}\right) = -1 \Leftrightarrow b \in \overline{Q}_p$.
3. Записать $p - 1 = 2^s t$, где t нечетно.
4. Найти $a^{-1} \mod p$ используя РАЭ.
5. Положить $c \leftarrow b^t \mod p$ и $r \leftarrow a^{(t+1)/2} \mod p$.
6. For $1 \leq i \leq s-1$ выполнить след.:
6.1 Вычислить $d = (r^2 \cdot a^{-1})^{2^{s-i-1}} \mod p$.
6.2 Если $d = -1 \mod p$, то положить $r \leftarrow r \cdot c \mod p$.
6.3 Положить $c \leftarrow c^2 \mod p$.
7. Return$(r, -r)$.

Алгоритм 1 при $s = 1$ сводится к алгоритму 2

Алгоритм 2 нахождения квадратных корней когда $p \equiv 3 \mod 4$
INPUT: нечетное простое число $p \equiv 3 \mod 4$ и целое число a: $1 \leq a \leq p-1$.
OUTPUT: два квадратных корня для a по модулю p, при условии, что $a \in Q_p$.
1. Вычислить символ Лежандра $\left(\frac{a}{p}\right) =: a^{\frac{p-1}{2}} \mod p$. Если $\left(\frac{a}{p}\right) = -1$, то a не имеет квадратных корней (т.е. $a \notin Q_p$) и остановиться.
2. Вычислить $r = a^{(p+1)/4} \mod p$.
3. Return$(r, -r)$.

Алгоритм 1 при $s = 2$ сводится к алгоритму 3
Алгоритм 3 нахождения квадратных корней когда $p \equiv 5 \mod 8$

INPUT: нечетное простое число $p \equiv 5 \mod 8$ и целое число a: $1 \leq a \leq p-1$.
OUTPUT: два квадратных корня для a по модулю p, при условии, что $a \in Q_p$.
1. Вычислить символ Лежандра $\left(\frac{a}{p}\right) =: a^{\frac{p-1}{2}} \mod p$. Если $\left(\frac{a}{p}\right) = -1$, то a не имеет квадратных корней (т.е. $a \notin Q_p$) и остановиться.
2. Вычислить $d = a^{(p-1)/4} \mod p$.
3. Если $d = 1$, тогда вычислить $r = a^{(p+3)/8} \mod p$.

4. Если $d = p - 1$, тогда вычислить
$r = 2a(4a)^{(p-5)/8} \mod p$.
5. Return$(r, -r)$.

Пример. $p = 331$, $a = 214$.
Найти $x : x^2 = 214 \mod 331$.

$331 = 3 \mod 4 \Rightarrow$ **Алгоритм 2**
$p - 1 = 330 = 2 \cdot 165$.
1.

$$\begin{aligned}\left(\frac{214}{331}\right) &= (214)^{\frac{331-1}{2}} \mod 331 = (214)^{165} \mod 331\\ &= ((214)^2)^{82}(214) \mod 331\\ &= (118)^{82}(214) \mod 331\\ &= ((118)^2)^{41}(214) \mod 331\\ &= (22)^{41}(214) \mod 331\\ &= ((22)^4)^{10}(22)(214) \mod 331\\ &= (239)^{10}(22)(214) \mod 331\\ &= ((239)^2)^5(22)(214) \mod 331\\ &= (189)^5(22)(214) \mod 331\\ &= ((189)^2)^2(189)(22)(214) \mod 331\\ &= (304)^2(189)(22)(214) \mod 331\\ &= (67)(189)(22)(214) \mod 331\\ &= (67)(189)(22)(214) \mod 331\\ &= 1 \mod 331 \Leftrightarrow 214 \in Q_{331}\end{aligned}$$

2.

$$\begin{aligned}r &= (214)^{\frac{331+1}{4}} \mod 331\\ &= (214)^{83} \mod 331\\ &= ((214)^2)^{41}(214) \mod 331\end{aligned}$$

$$\begin{aligned}&= (118)^{41}(214) \mod 331\\ &= ((118)^2)^{20}(118)(214) \mod 331\\ &= (22)^{20}(118)(214) \mod 331\\ &= ((22)^4)^5(118)(214) \mod 331\\ &= (239)^5(118)(214) \mod 331\end{aligned}$$

$$= ((239)^2)^2(239)(118)(214) \mod 331$$
$$= (189)^2(239)(118)(214) \mod 331$$
$$= (304)(239)(118)(214) \mod 331$$
$$= 144 \mod 331$$

3. Т.О Корни $r = 144$ и $r = -144 = 187 \mod 331$

Пример. $p = 277$, $a = 63$.
Найти $x : x^2 = 63 \mod 277$.
$277 = 5 \mod 8 \Rightarrow$ **Алгоритм 3**
$p - 1 = 276 = 2^2 \cdot 3 \cdot 23$.

1.

$$\left(\frac{63}{277}\right) = 63^{\frac{277-1}{2}} \mod 277 = 63^{138} \mod 277$$
$$= ((63)^2)^{69} \mod 277 = (91)^{69} \mod 277$$
$$= ((91)^2)^{34}(91) \mod 277 = ((248)^2)^{17}(91) \mod 277$$
$$= (10)^{17}(91) \mod 277$$
$$= (((10)^2)^2)^4(10)(91) \mod 277$$
$$= (28)^4(10)(91) \mod 277$$
$$= (270)(10)(91) \mod 277$$
$$= 1 \mod 277.$$

2.

$$d = 63^{\frac{277-1}{4}} \mod 277$$
$$= 63^{69} \mod 277$$
$$= ((63)^2)^{34}(63) \mod 277$$
$$= (91)^{34}(63) \mod 277$$
$$= ((91)^2)^{17}(63) \mod 277$$
$$= (248)^{17}(63) \mod 277$$
$$= ((248)^2)^8(248)(63) \mod 277$$
$$= (10)^8(248)(63) \mod 277$$
$$= (((10)^2)^2)^2(248)(63) \mod 277$$
$$= (28)^2(248)(63) \mod 277$$
$$= (230)(248)(63) \mod 277$$
$$= 276 \mod 277$$
$$= p - 1 \mod 277.$$

4.

$$\begin{aligned} r &= 2(63)(4\cdot 63)^{\frac{277-5}{8}} \bmod 277 \\ &= 2(63)(4\cdot 63)^{34} \bmod 277 \\ &= 2(63)((252)^2)^{17} \bmod 277 \\ &= 2(63)(71)^{17} \bmod 277 \\ &= 2(63)(71)((71)^2)^8 \bmod 277 \\ &= 2(63)(71)(55)^8 \bmod 277 \\ &= 2(63)(71)((55)^2)^4 \bmod 277 \\ &= 2(63)(71)(255)^4 \bmod 277 \\ &= 2(63)(71)((255)^2)^2 \bmod 277 \\ &= 2(63)(71)(207)^2 \bmod 277 \\ &= 2(63)(71)(191) \bmod 277 \\ &= 150 \bmod 277 \end{aligned}$$

5. Т.О Корни $r = 150$ и $r = -150 = 127 \bmod 277$

Теорема 2 о квадратичных вычетах и квадратных корнях по модулю составного числа $n = pq$, где p, q — различные нечетные простые числа.
(i) $|Q_p| = \frac{\varphi(n)}{4} = \frac{(p-1)(q-1)}{4}$. В частности,

$$\left|\overline{Q}_p\right| = \frac{3\varphi(n)}{4} = \frac{3(p-1)(q-1)}{4}. \tag{6}$$

(ii) Если $a \in Q_n$, то a имеет в $\mathbb{Z}_n^*$ 4 различных квадратных корня.
(iii) Имеем

$$a \in Q_n \;\Leftrightarrow\; a \bmod p \in Q_p \quad \text{и} \quad a \bmod q \in Q_q. \tag{7}$$

Алгоритм 4 нахождения квадратных корней для a по модулю составного числа $n = pq$, где p, q — различные нечетные простые числа.

INPUT: целое число n с простыми факторами p, q и $a \in Q_n$.

OUTPUT: 4 квадратных корня для a по модулю n.

1. С помощью алг.1 (или алг.2 или алг.3 если они применимы) найти два квадратных корня r, $-r$ для a по модулю p.
2. С помощью алг.1 (или алг.2 или алг.3 если они применимы) найти два квадратных корня s, $-s$ для a по модулю q.

3. С помощью РАЕ найти $c, d \in \mathbb{Z}$: $cp + dq = 1$.
4. Положить $x \leftarrow (rdq + scp) \mod n$ и
$y \leftarrow (rdq - scp) \mod n$.
5. Return($\pm x \mod n, \pm y \mod n$).

Пример (продолжение). $p = 331$, $q = 277$, $n = pq = 91687$,
$a = 62111$. Найти $x : x^2 = 62111 \mod 91687$.
1. $62111 = 214 \mod 331 \Rightarrow$ (см. прим.4)
$r = 114 \mod 331$, $-r = 187 \mod 331$
2. $62111 = 63 \mod 277 \Rightarrow$ (см. прим.5)
$s = 150 \mod 277$, $-s = 127 \mod 277$
3. $c = 118$, $d = -141$
4. $x = (rdq + scp) \mod n = 51118 \Rightarrow$
$r_1 = x = 51118$
$r_2 = -x \mod 91687 = 40569$.
$y = (rdq - scp) \mod n = 69654 \Rightarrow$
$r_3 = y = 69654$
$r_4 = -y \mod 91687 = 22033$.
5. 4 квадратных корня
$r_1 = x = 51118$, $r_2 = 40569$, $r_3 = 69654$, $r_4 = 22033$.

ЛАБОРАТОРНАЯ РАБОТА 4

Алгоритмы сортировки

1) Сортировка вставками

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2        do key ← A[j]
3           ▷ добавить A[j] к отсортированной части A[1 . . j − 1].
4           i ← j − 1
5           while i > 0 and A[i] > key
6                 do A[i + 1] ← A[i]
7                    i ← i − 1
8           A[i + 1] ← key
```

2) Сортировка слиянием

MERGE(A, p, q, r)

```
 1  n1 ← q − p + 1
 2  n2 ← r − q
 3  Создаем массивы L[1..n1 + 1] и R[1..n2 + 1]
 4  for i ← 1 to n1
 5        do L[i] ← A[p + i − 1]
 6  for j ← 1 to n2
 7        do R[j] ← A[q + j]
 8  L[n1 + 1] ← ∞
 9  R[n2 + 1] ← ∞
10  i ← 1
11  j ← 1
12  for k ← p to r
13        do if L[i] ≤ R[j]
14              then A[k] ← L[i]
15                   i ← i + 1
16              else A[k] ← R[j]
17                   j ← j + 1
```

MERGE-SORT(A, p, r)

```
1  if p < r
2     then q ← ⌊(p + r)/2⌋
3          MERGE-SORT(A, p, q)
4          MERGE-SORT(A, q + 1, r)
5          MERGE(A, p, q, r)
```

Описание

1. вычисляется длина n_1 подмассива $A[p..q]$,

2. вычисляется длина n_2 подмассива $A[q+1..r]$,

3. создаются левый и правый массивы, длины которых равны n_1+1, n_2+1 соответственно.

4-5) подмассив $A[p..q]$ копируется в массив $L[1..n_1]$,

6-7) подмассив $A[q+1..r]$ копируется в массив $R[1..n_2]$,

8-9) последним элементам массивов L, R приписываются сигнальные значения.