תרגול באלגוריתמים 28.12.2004

ייצוג בינרי:

מילה N, אורכה יהיה:

$$n = \lceil \log(n) \rceil.$$

אם נתון N = p.q, אז האלגוריתם הנאיבי למציאת ייצוג בינרי יהיה:

```
1    for i = 2 to N-1
2        for j = 2 to N-1
3            if i.j = N
4                return i,j
```

זמן הריצה: $O(2^{2n}) = O(N^2)$

כש- n הוא אורך הייצוג.

הגדרות:

d|n מציין ש d>0 מחלק את n.

כלומר שקיים k ∈ Z כך ש n = k.d.

a|b: |a| <= |b| או b = 0.

משפט:

לכל a,n ∈ Z קיימים q,r יחידים כך ש- 0 <= r < n : a = q.n + r ויקרא הפירוק של a לפי n, ויתקיים:

- $q = \lfloor a/n \rfloor$.
- r = a mod n.

טענה:

אם $d|a$ ו $d|b$ אזי $d|ax+by$,$x,y \in Z$.

הוכחה:

$a=k_1.d$ ו $b=k_2.d$, **נציב**:

$ax+by = k_1.dx+k_2.by = d(k_1.x+k_2.y)$ וסיימנו.

Gcd(a,b): המחלק משותף המקסימלי.

דוגמאות:

- $Gcd(30,24) = 6$
- $Gcd(0,4) = 4$
- הגדרה: $Gcd(0,0) = 0$

משפט:

$Gcd(a,b)$ הוא המינימום החיובי ממש של הקבוצה $\{ ax+by, x,y \in Z \}$

כלומר קבוצת הצירופים הליניאריים של a,b.

הוכחה:

יהי $S>0$ המספר המינימלי הזה.

- כיוון ראשון:

  נסתכל על פירוק a לפי S: $a = qs + r$ אז:

  $r = a – qs = a – q(ax + by) = a(1-qx) – bqy$

  מהגדרת הפירוק $0 <= r < s$ וממינימליות S חייב להתקיים $r = 0$.

  מסקנה: $S|a$ כנ"ל ל $S|b$.

  לכן בחלק הראשון הראנו ש: $Gcd(a,b) >= S$.

- כיוון שני:

  S הוא צירוף ליניארי של a,b לכן $Gcd(a,b)|S$ לכן $Gcd(a,b) <= S$.

מסקנה: Gcd(a,b) הוא צירוף ליניארי של a,b לכן כל מחלק משותף d יקיים d|Gcd(a,b).

משפט:

Gcd(a,b) = Gcd(b, a mod b)

הוכחה:

- כיוון ראשון:

  נראה ש:

  Gcd(a,b)|Gcd(b,amodb)

  יהי d = Gcd(a,b).

  נפרק את a לפי b:

  a = qb + a mod b

  אזי a mod b = a-qb

  כלומר a mod b הוא צירוף ליניארי של a,b לכן d|a mod b.

  מסקנה: d|Gcd(b,a mod b)

- כיוון שני:

  d` = Gcd(b, a mod b)

  a = qb + a mod b לכן d`|a ומכאן d`|Gcd(a,b).

האלגוריתם לחישוב Gcd(a,b):

Euclid(a, b):

```
1    if b = 0
2        return a
3    else
4        Euclid( b, amodb )
```

אפשר להניח ש- a > b, כי:

אם a > b: אז ברור שהאלגוריתם עובד.

אם a = b: אז האלגוריתם עוצר אחרי שתי איטרציות, ונותן את התוצאה הנכונה.

אם a < b: אז באיטרציה השניה כבר רואים שמחושב Gcd(b,a), כלומר האלגוריתם מחליף בין a ל b.

נכונות האלגוריתם:

רואים שהאלגוריתם תמיד עוצר והשאר נובע ישר מהמשפט שהוכחנו.

דוגמא:

Euclid(30,24) = Euclid(24,6) = Euclid(6,0) = 6

זמן הריצה:

טענה: נניח ש Euclid(a,b) קורא ריקורסיבית ל Euclid(x,y), אזי y <= a/2.

הוכחה:

- מקרה (1):

  a >= 2b אזי: y = a mod b < b <= a/2.

- מקרה (2):

  B < a < 2b <== a < 2(a-b).

  y = a mod b = a-b < a/2.

מסקנה: בכל שתי קריאות, שני הערכים קטנים פי 2 לפחות.

לכן, מספר הקריאות הרקורסיביות O(m) = O(log(b)).

בכל קריאת מבצעים חלוקות והכפלות: $O(m^2)$.

לכן בסה"כ נקבל: $\mathbf{O(m^3)}$.

נרחיב את Euclid כך שהפלט שלו יהיה (d,x,y) כך ש- d = ax + by:

Extended-Euclid(a, b):

```
1    if b = 0
2        return (a,1,0)
3    else
4        (d`,x`,y`) ← Euclid( b, a mod b )
5        return (d`,y`,x`-qy`)
```

כאשר q הוא מ: a = qb + c כלומר:

$q=a/\lfloor b \rfloor$.

נכונות האלגוריתם:

- $d = d`$ ראינו.
- בשורה הרביעית יש בעצם אינדוקציה.

  $d = d` = x`b + y`.a \bmod b$

  לפי הנחת האינדוקציה:

  $x`b + y`amodb = x`b + y`(a-qb) = y`a + (x`-qy`)b$

  כלומר d צירוף ליניארי של $y`, x`-qy`$ וזה מה שהראינו.

זמן ריצה:

לא משתנה כלומר שווה ל: $\mathbf{O(m^3)}$.

משוואות מודלריות:

בעיה : רוצים לפתור $ax \equiv b(modn)$

דוגמאות:

- $2x \equiv 2(mod4)$

  $x = 1 \text{ or } 3$

- $2x \equiv 1(mod6)$

כלומר קיימים x,h כך ש- $2x \equiv 6h + 1$ אבל הצד השמאלי זוגי והימני אי-זוגי כלומר אין פיתרון.

משפט:

ל- $ax \equiv b(modn)$ יש פיתרון $\Leftrightarrow Gcd(a,n)|b$.

הוכחה:

$va + un = d = Gcd(a,n)$

<u>==>:</u>

קיימים x,h כך ש:

b = ax - nb <== ax = nh + b

לכן b הוא צירוף ליניארי של a,n ולכן: d|b.

<u><==:</u>

קיים k כך ש: b = k.d = k.(av+un)

האם קיימים x, h המקיימים: ax = nh + k(av+un)?

קובעים: x = kv, h = -ku ורואים שאכן הם מקיימים לכן יש פתרון למשוואה.

<u>אלגוריתם למציאת פתרון כלשהו ל – ax = b(mod n):</u>

נריץ את Extended-Euclid(a,n) ונקבל בחזרה (d,v,u) עכשיו נחלק את b ב- d ונקבל את k.

הפלט יהיה: kv והוא פיתרון המשוואה.

<u>דוגמא:</u> $2x = 2 \pmod 4$:

Extended-Euclid מחזיר (2,-1,1) כש d=2, v=-1 ו u=1, אז k=1, לכן 1- פתרון של המשוואה וב-mod4 זה שווה ל- 3.

<u>העלאה בחזקה:</u>

נתונים a,b,n רוצים לחשב את: $a^b \pmod n$.

האלגוריתם:

$b = \langle b_k, b_{k-1}, .., b_0 \rangle$ הייצוג הבינארי של b כש b_k הוא ה- MSB אזי:

<u>Exponent (a,b,n)</u>

```
1    d ← 1
2    for i = k down to 0
3        d ← d^2(mod n)
4        if b_i = 1
5            d ← d.a(mod n)
6    return d
```

נכונות האלגוריתם:

דוגמא:

$a^{10} \pmod n$:

b = 10 (עשרוני) = 1010 (בינרי) = $1.2^3 + 0.2^2 + 1.2^1 + 0.2^0$ (עשרוני)

$Ab = 1.(a^2)^3.a^0.(a^2)^1.a0$

באלגוריתם:

$(((1^2.a)^2)^2.a)^2 \ldots = (a^2)^3.(a^2)^1$

זמן ריצה:

נניח ב- Exponent ש- $|a| = |b| = |n| = k$

אזי זמן הריצה יהי $\mathbf{O(k^3)}$.

חבורות:

חבורה (S, □):

S: קבוצה של איברים.

□: כפל, פעולה בינרית.

דרישות:

1. $\forall a,b \in S : a \square b \in S$.
2. אסוציאטיביות: $\forall a,b \in S$:

   $(a \square b) \square c = a \square (b \square c)$
3. איבר יחידה: קיים $e \in S$ יחיד כך ש: $e \square a = a \square e = a$
4. קיום הפכי: לכל $a \in S$ קיים הפכי יחיד a^{-1} כך ש $a \square a^{-1} = a^{-1} \square a = e$

<u>דוגמאות:</u>

1. לכל $n \in N$: $(Z_n, +)$

$Z_n = \{0<= a <=n-1\}$

למשל: $Z_6 = \{0,1,2,3,4,5\}$

הפעולה +: חיבור מודלו n.

$4 + 5 = 3$.

איבר היחידה: 0.

הופכי: $3 = 3^{-1}$.

2. לכל $n \in N$: $(Z_n^*, .)$

$Z_n^* = \{0< a <=n-1 : Gcd(a,n) = 1 \}$

למשל: $Z_6^* = \{1, 5\}$

הפעולה . : כפל מודלו n.

איבר היחיד: 1.

הופכי: $5 = 5^{-1}$ (כל איבר הופכי של עצמו) .

גם כן: $Z_{12}^* = \{1,5,7,11\}$

הוכן ע"י: ודיע ג'מל