Projektovanje algoritama

L09. Algoritmi iz teorije brojeva. Kriptografija

Današnje teme

- Algoritmi sa matricama
- Algoritmi iz teorije brojeva
- Algoritmi calculus-a

Množenje matrica

```
SQUARE-MATRIX-MULTIPLY(A,B)
  n = A.rows
  let C be a new n x n matrix
  for i = 1 to n
    for j = 1 to n
      c[i,j] = 0
      for k = 1 to n
        c[i,j] = c[i,j] + a[i,k]*b[k,j]
  return C
```

$$T(n) = \theta(n^3)$$

Množenje matrica – rekurzivni pristup

```
MAT-MUL-REC(A,B)
  n = A.rows
  let C be a new n x n matrix
  if n == 1
    c[1,1] = a[1,1] * b[1,1]
  else partition A,B,C into n/2 x n/2 matrices
    C[1,1] = MAT-MUL-REC(A11,B11) + MAT-MUL-REC(A12,B21)
    C[1,2] = MAT-MUL-REC(A11,B12) + MAT-MUL-REC(A12,B22)
    C[2,1] = MAT-MUL-REC(A21,B11) + MAT-MUL-REC(A22,B21)
    C[2,2] = MAT-MUL-REC(A21,B12) + MAT-MUL-REC(A22,B22)
  return C
```

$$T(n) = \theta(n^3)$$

Množenje matrica – Strassen metoda [1]

```
S[1] = B[1,2] - B[2,2]
S[2] = A[1,1] + A[1,2]
S[3] = A[2,1] + A[2,2]
S[4] = B[2,1] - B[1,1]
S[5] = A[1,1] + A[2,2]
S[6] = B[1,1] + B[2,2]
S[7] = A[1,2] - A[2,2]
S[8] = B[2,1] + B[2,2]
S[9] = A[1,1] - A[2,1]
S[10] = B[1,1] + B[1,2]
```

$$T(n) = \theta(n^2)$$

Množenje matrica – Strassen metoda [2]

```
P[1] = A[1,1] * S[1]
P[2] = S[2] * B[2,2]
P[3] = S[3] * B[1,1]
P[4] = A[2,2] * S[4]
P[5] = S[5] * S[6]
P[6] = S[7] * S[8]
P[7] = S[9] * S[10]
```

$$T(n) = 7T\left(\frac{n}{2}\right)$$

Množenje matrica – Strassen metoda [3]

```
C[1,1] = P[5] + P[4] – P[2] + P[6]
C[1,2] = P[1] + P[2]
C[2,1] = P[3] + P[4]
C[2,2] = P[5] + P[1] – P[3] – P[7]
```

$$\boldsymbol{T(n) = \theta\left(n^{\lg 7}\right)}$$

Nalaženje najvećeg zajedničkog delioca

```
EUCLID-GCD(a,b)
  if b == 0
    return a
  else
    return EUCLID-GCD(b, a mod b)
```

$$T(n) = \theta(\lg b)$$

Kriptografski sistemi sa javnim ključem

- Primena:
  - Komunikacija između dve tačke bez mogućnosti prisluškivanja
  - Digitalni potpisi bez mogućnosti krivotvorenja
- Koristi osobine prostih brojeva:
  - Pronaći veliki prost broj – lako
  - Faktorisati proizvod dva velika prosta broja – teško

Kriptografski sistemi sa javnim ključem

- Dva tipa ključa:
  - Javni ključ (*Public Key*): P_A, P_B
  - Tajni ključ (*Secret Key*): S_A, S_B
- Tajni ključ se ni sa kim ne deli, a javni ključ je javno dostupan.
- Svaki učesnik u komunikaciji formira svoj javni i svoj tajni ključ.

- Važan uslov: Svaki ključ je 1-1 funkcija unutar skupa poruka!
  - Vremenski uslov: Funkcija koju definiše ključ je efikasno izračunljiva.

Kriptografski sistemi sa javnim ključem

- Za svakog učesnika u komunikaciji mora da važi:

$$S_A\big(P_A(M)\big) = M$$

$$P_A\big(S_A(M)\big) = M$$

- Iako je P_A javno dostupna, S_A ne sme biti laka za otkriti!
- Javni ključ služi za **enkripciju**, a tajni ključ služi za **dekripciju**.
- Za digitalni potpis, tajnim ključem se **potpisuje**, a javnim **proverava** autentičnost potpisa.

RSA sistem sa javnim ključem

1. Izabrati dva velika prosta broja p i q (danas: 768 – 2048 bit)
2. Izračunati $n = p * q$
3. Izabrati mali neparan broj e relativno prost sa $(p-1) * (q-1)$
4. Izračunati broj d takav da je $d * e \equiv 1 \left(mod\big((p-1) * (q-1)\big)\right)$
5. Javni ključ: $P = (e, n)$ $\quad P(M) = M^e \ (mod\ n)$
6. Tajni ključ: $S = (d, n)$ $\quad S(M) = M^d \ (mod\ n)$

RSA sistem sa javnim ključem - primer

$p = 11$
$q = 29$
$n = 319$
$e = 3$

RSA sistem sa javnim ključem - primer

$p = 11$
$q = 29$
$n = 319$
$e = 3$

$d * e \equiv 1 \ (mod \ 10 * 28) \ \rightarrow 3d = 280k + 1 \ \rightarrow d = 187$

(do vrednosti broja d možemo doći pokušajem definisanja vrednosti za k od 1 pa naviše, dok ne dobijemo celi broj d)

$P = (3, 319)$
$S = (187, 319)$