# Algorithms and data structures 4: Matricies-2

Boris Kirikov

15.10.2015

# Outline

- Revision
- Understanding matricies
- Gauss algo

# Revision

- What is matrix?
- Operations:
  - Sum
  - Multiply by scalar
  - Multiplication
  - Transpose

# Understanding matricies

- Mapping $I \times J \to R$
- Table with numbers
- Graph

# Understanding matricies

- Mapping $I \times J \to R$
- Table with numbers
- Graph

- System of linear equations
- Linear operator
- Bilinear form
- Automata and Markov chain

# Understanding matrisies: linear space

- $V_R$ — linear space over $R$ — scalars
- Has operations: sum, multiply by scalar
- Linear closure: $< v^{(1)}, \ldots, v^{(n)} >= \{\alpha_1 v^{(1)} 1 + \ldots + \alpha_n v^{(n)}, \alpha_i \in R\} \subset V_R$
- Basis — minimal $< v_1, \ldots, v_n >= V_R$
- dim $V = n$ if $< v_1, \ldots, v_n >= V_R$

# Understanding matrisies: linear space

- $V_R$ — linear space over $R$ — scalars
- Has operations: sum, multiply by scalar
- Linear closure: $< v^{(1)}, \dots, v^{(n)} >= \{\alpha_1 v^{(1)} 1 + \ldots + \alpha_n v^{(n)}, \alpha_i \in R\} \subset V_R$
- Basis — minimal $< v_1, \dots, v_n >= V_R$
- dim $V = n$ if $< v_1, \dots, v_n >= V_R$

**Ex**: $R = \mathbb{R}, V = \mathbb{R}^n$. Vectors are $v = (v_1, \dots, v_n)$. Standard basis:

$$e_1 = (1, 0, \dots, 0)$$

$$e_2 = (0, 1, \dots, 0)$$

$$e_n = (0, \dots, 0, 1)$$

$$e_i^{(j)} = \delta_{ij}$$

# Understanding matricies: linear mapping

**Def**: $f : X_R \to Y_R$ is linear, iff $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$.

**Th**: Let $L(X, Y)$ be a set of all linear mappings $X_R \to Y_R$, $n = \dim X$, $m = \dim Y$. Than $L(X, Y) = M(n, m, R)$ (up to isomorphism).

But what means linear mapping in linear space?

# Understanding matricies: system of linear equations

Kelly's definition of multiplication lets us to write system of linear equations in matrix form:

$$\begin{cases} a_{11}x_1 & + & \dots & + & a_{1n}x_n & = & b_1 \\ & + & \dots & + & & = & \\ a_{m1}x_1 & + & \dots & + & a_{mn}x_n & = & b_m \end{cases}$$

$$Ax = b$$

How do we solve such systems?

# Gauss algo: step 1

- Staircase form

## Gauss algo: step 1

- Staircase form
- What operations we performed?

# Gauss algo: step 2

- Eliminating variables
- What operations we performed?

# Gauss algo and matricies: $LU$-decomposition

- Step 1 of Gauss algo gives (under some constraints) triangular matrix
- Operations performed during this step also can be represented with triangular matrix
- This is $A = LU$, $L$ — matrix of operations, $U$ — simplified $A$

# EOF

*We were not very fast, so other material will be on the next lecture.*