Problems and solutions from entrance exam in Yandex school of data analysis

Nemesh N. T.

1 Interview questions

1.1 Algebra

Problem 1.1.1 *Count all bases of n-dimensional vector space over field $\mathbb{F}_k$.*

◁ There are $k^n - 1$ ways to choose the first vector, Now there are $k^n - k$ ways to choose the second linearly independent vector. Then there are $k^n - k^2$ ways to choose the third linearly independent vector and etc. Hence there are

$$(k^n - 1)(k^n - k)(k^n - k^2) \cdot \ldots \cdot (k^n - k^{n-1})$$

ways to choose n linearly independent vectors in $(\mathbb{F}_k)^n$, i.e. to construct a basis. ▷

Problem 1.1.2 *Find the probability that a random matrix of size $n \times n$ with entries in the finite field $\mathbb{F}_k$ will be non-singular.*

◁ The total number of matrices of size $n \times n$ with entries in $\mathbb{F}_k$ is k^{n^2}. Now we want to compute the number of non-singular matrices. Rows of these matrices are linearly independent. There are $k^n - 1$ ways to choose the first row, i.e. the first vector. Now there are $k^n - k$ ways to choose the second vector. Then there are $k^n - k^2$ ways to choose the third vector and etc. Hence there are $(k^n - 1) \cdot \ldots \cdot (k^n - k^{n-1})$ ways to choose n linearly independent rows. This is the number of non-singular matrices. Hence the probability is

$$\frac{(k^n - 1) \cdot \ldots \cdot (k^n - k^{n-1})}{k^{n^2}}$$

▷

Problem 1.1.3 *Find all $f \in \mathbb{R}[x]$ such that $xf(x-2) = (x-2)f(x-1)$*

◁ After substitution $x = 2$ we get $2f(0) = 0$, i.e. $f(0) = 0$. This means that $f(x) = xg(x)$ for some $g \in \mathbb{R}[x]$. In this case our equation can be rewritten as $x(x-2)g(x-2) = (x-2)(x-1)g(x-1)$, which is equivalent to $xg(x-2) = (x-1)g(x-1)$. After substitution $x = 1$ we get $g(-1) = 0$, hence $g(x) = (x+1)h(x)$ for some $h \in \mathbb{R}[x]$. Now the last equation can be rewritten as $x(x-1)h(x-2) = (x-1)xh(x-1)$, i.e. $h(x-2) = h(x-1)$. In particular $h(\mathbb{Z}) = \{h(0)\}$. The only polynomial attaining the same value infinitely many times is the constant polynomial, i.e. $h(x) = C$. Finally, $f(x) = xg(x) = x(x+1)h(x) = Cx(x+1)$. ▷

Problem 1.1.4 *Find all $f \in \mathbb{R}[x]$ such that $(x-3)f(x-1) = (x-2)f(x)$*

◁ Substitute $x = 3$ to get $f(3) = 0$, hence $f(x) = (x-3)f_1(x)$ for some $f_1 \in \mathbb{R}[x]$. Then our equation can be rewritten as $(x-3)(x-4)f_1(x-1) = (x-2)(x-3)f_1(x)$, so $(x-4)f_1(x-1) = (x-2)f_1(x)$. Again substitute $x = 4$ to get $f_1(4) = 0$, hence $f_1(x) = (x-4)f_2(x)$ for some $f_2 \in \mathbb{R}[x]$. We can repeat this process and for any $n \in \mathbb{N}$ to construct a polynomial $f_n \in \mathbb{R}[x]$ such that $(x-3-n)f_n(x-1) = (x-2)f_n(x)$ and $f_{n-1}(x) = (x-n-2)f_n(x)$. In this case we have $f(x) = (x-3)(x-4) \cdot \ldots \cdot (x-n-2)f_n(x)$ for all $n \in \mathbb{N}$. This is possible iff $f(x) = 0$. ▷

1.2 Linear algebra

Problem 1.2.1 *Given a Jordan matrix of an operator count all its invariant subspaces*

◁ By Jordan's decomposition

$$\mathcal{A} = \bigoplus_{i=1}^{m} \mathcal{A}_i \qquad \mathcal{A}_i = \bigoplus_{l=1}^{m_i} \mathcal{J}_{n_i,l}(\lambda_i)$$

where $(\lambda_i)_{i\in\mathbb{N}_m}$ are eigenvalues of $\mathcal{A}$ and

$$V_{\lambda_i}(\mathcal{A}) := \operatorname{Ker}(\mathcal{A} - \lambda_i\mathcal{E}) \qquad k_i := \dim V_{\lambda_i}(\mathcal{A})$$

If $k_i > 1$ for some $i \in \mathbb{N}_m$, then consider linearly independent vectors $e_1, e_2 \in V_{\lambda_i}(\mathcal{A})$. Clearly, $\operatorname{span}\{e_1\}$, $\operatorname{span}\{e_2\}$ are an invariant subspaces of $\mathcal{A}$. For all $\mu_1, \mu_2 \in \mathbb{C}$ we see that $\operatorname{span}\{\mu_1 e_1 + \mu_2 e_2\}$ is an invariant subspace of $\mathcal{A}$ too. Hence we have infinitely many invariant subspaces.

If $k_i = 1$ for all $i \in \mathbb{N}_m$, then $\mathcal{A}_i = \mathcal{J}_{n_i}(\lambda_i)$. For each $i \in \mathbb{N}_m$ the subspaces $U_{i,r} = \operatorname{Ker}(\mathcal{A} - \lambda_i\mathcal{E})^r$ with $r \in \mathbb{N}^*_{n_i}$ are an increasing chain of invariant subspaces. An arbitrary invariant subspace is of the form

$$\bigoplus_{i=1}^{m} U_{i,r_i}$$

where $r_i \in \mathbb{N}^*_{n_i}$. Consequently, th number of invariant subspaces is

$$\prod_{i=1}^{m} (n_i + 1)$$

▷

Problem 1.2.2 *Give an example of a linear operator with only one eigenvector*

◁ Consider Jordan's matrix $\mathcal{J}_n(\lambda)$. Since

$$\det(\mathcal{J}_n(\lambda) - t\mathcal{E}) = (\lambda - t)^n$$

then we have only one eigenvalue λ with multiplicity n. Now we define eigenvectors from the equation

$$(\mathcal{J}_n(\lambda) - \lambda\mathcal{E})x = 0$$

This is equivalent to $\mathcal{J}_n(0)x = 0$. It is easily seen that solutions of this equation are of the form

$$x = (c, 0, 0, \ldots, 0)^{tr} \qquad c \in \mathbb{C}$$

Hence $\mathcal{J}_n(\lambda)$ has only one eigenvector. ▷

Problem 1.2.3 *Give an example of a linear operator without eigenvalues, but with proper innvariant subspaces.*

◁ By $\mathcal{R}(\alpha)$ we denote the linear operator corresponding to rotation by angle α. Its matrix is

$$[\mathcal{R}(\alpha)] = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

Now we find

$$\det(\mathcal{R}(\alpha) - t\mathcal{E}) = (t - \cos\alpha)^2 + \sin^2\alpha$$

Which is strictly greater than zero provided $\alpha \notin \pi\mathbb{Z}$. Now consider angles $(\alpha_i)_{i\in\mathbb{N}_m} \subset \mathbb{R} \setminus \pi\mathbb{Z}$. We define a linear operator

$$\mathcal{R} = \bigoplus_{i=1}^{m} \mathcal{R}(\alpha_i)$$

Since $[\mathcal{R}]$ is a block-diagonal matrix, then $\mathcal{R}$ admits invariant subspaces. In fact it has at least 2^m invariant subspaces. But $\mathcal{R}$ has no eigenvalues. Indeed

$$\det(\mathcal{R} - t\mathcal{E}) = \prod_{i=1}^{m} \det(\mathcal{R}(\alpha_i) - t\mathcal{E}) = \prod_{i=1}^{m}((t - \cos\alpha_i)^2 + \sin^2\alpha_i) > 0$$

because of the choice of the angles $(\alpha_i)_{i\in\mathbb{N}_m}$. ▷

Problem 1.2.4 *Compute $\mathcal{J}_n(\lambda)^k$ for all $k \in \mathbb{N}$.*

◁ Note that for commuting operators $\mathcal{A}$ and $\mathcal{B}$ and for all $k \in \mathbb{N}$ we have the binomial identity

$$(\mathcal{A} + \mathcal{B})^k = \sum_{s=0}^{k} \binom{k}{s} \mathcal{A}^s \mathcal{B}^{k-s}$$

Hence,

$$\mathcal{J}_n(\lambda)^k = (\mathcal{J}_n(0) + \lambda\mathcal{E})^k = \sum_{s=0}^{k} \binom{k}{s} \mathcal{J}_n(0)^s (\lambda\mathcal{E})^{k-s} = \sum_{s=0}^{k} \binom{k}{s} \lambda^{k-s} \mathcal{J}_n(0)^s$$

Now it remains to note that $[\mathcal{J}_n(0)^s]_{i,j} = \delta_{i,j-s}$ for all $s \in \mathbb{N}^*$. We can compute arbitrary entry of $\mathcal{J}_n(\lambda)^k$. Indeed,

$$[\mathcal{J}_n(\lambda)^k]_{i,j} = \sum_{s=0}^{k} \binom{k}{s} \lambda^{k-s} \delta_{i,j-s} = \binom{k}{j-i} \lambda^{k-(j-i)}$$

For example

$$[\mathcal{J}_3(\lambda)^4] = \begin{pmatrix} \binom{4}{0}\lambda^4 & \binom{4}{1}\lambda^3 & \binom{4}{2}\lambda^2 \\ \binom{4}{-1}\lambda^5 & \binom{4}{0}\lambda^4 & \binom{4}{1}\lambda^3 \\ \binom{4}{-2}\lambda^6 & \binom{4}{-1}\lambda^5 & \binom{4}{0}\lambda^4 \end{pmatrix} = \begin{pmatrix} \lambda^4 & 4\lambda^3 & 6\lambda^2 \\ 0 & \lambda^4 & 4\lambda^3 \\ 0 & 0 & \lambda^4 \end{pmatrix}$$

▷

Problem 1.2.5 *Which eigenvalues can a matrix A satisfying $A^4 = A^2$ have? Give an example of matrix which have all these eigenvalues.*

◁ If λ is an eigenvalue of a matrix A, then $Ax = \lambda x$ for some $x \neq 0$. Hence, $A^k x = \lambda^k x$, Since $A^4 = A^2$, then $\lambda^4 x = \lambda^2 x$. Since $x \neq 0$, then $\lambda^4 = \lambda^2$. This equation has three roots: 0 and ± 1. An example of matrix which has all these eigenvalues is

$$A = \operatorname{diag}(0, -1, 1)$$

▷

Problem 1.2.6 *Given two symmetric* 2×2 *matrices determine whether they specify the same quadratic form.*

◁ In general we must compute positive and negative indices of inertia of matrices. They are the same if and only if these matrices describe the same quadratic form. In order to compute indices of inertia we must find eigenvalues of the matrices. Number of positive/negative eigenavalues are positive/negative indices of inertia respectively. ▷

Problem 1.2.7 *Given two* 2×2 *matrices determine whether they specify the same linear operator.*

◁ In general we must compare Jordan's forms of given matrices.

In this particular case it is enough to say if matrices are different. If matrices describe the same linear operator, then they have the same characteristic polynomial. In case of 2×2 matrices we have

$$\det(\mathcal{A} - t\mathcal{E}) = \det(\mathcal{A}) - \operatorname{tr}(\mathcal{A})t + t^2$$

Hence if matrices have different determinants or traces, then the specify different linear operators. In general this method can't be used to distinguish matrices. For example the matrices

$$A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \qquad A_2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

have the same trace and determinant, but they specify different linear operators because

$$\dim(\operatorname{Ker}(\mathcal{A}_1)) = 2 \qquad \dim(\operatorname{Ker}(\mathcal{A}_2)) = 1$$

▷

1.3 Analysis

Problem 1.3.1 *Let $f\colon [0,1] \to [0,1]$ be a continuous map. Prove it has a fixed point.*

◁ Consider continuous function $g(x) = f(x) - x$. Obviously,

$$g(0) = f(0) - 0 = f(0) \in [0,1] \qquad g(1) = f(1) - 1 \in [-1,0]$$

Since $g(0) \geq 0$, $g(1) \leq 0$ by mean value theorem there exists $x \in [0,1]$ such that $g(x) = 0$. This means that $f(x) = x$, i.e. x is a fixed point of f. ▷

Problem 1.3.2 *Find derivative and domain of the function $f(x) = (\sin x)^{\cos x}$*

◁ Obviously,

$$\operatorname{dom}(f) = \{x \in \mathbb{R} : \sin x \geq 0\} = \bigcup_{n\in\mathbb{Z}} \{x \in \mathbb{R} : 2\pi n \leq x \leq \pi + 2\pi n\}$$

Now note that $\ln f(x) = \cos x \ln \sin x$. After taking the derivative we get

$$\frac{f'(x)}{f(x)} = (\cos x)' \ln(\sin x) + \cos x (\ln(\sin x))'$$

$$f'(x) = (\sin x)^{\cos x} (\cos x \cot x - \sin x \ln(\sin x))$$

▷

Problem 1.3.3 *Compute Maclaurin series of* $\arctan x$

◁ Note that for all $t \in (-1,1)$ we have

$$\frac{1}{1+t^2} = \sum_{k=0}^{\infty} (-t^2)^k = \sum_{k=0}^{\infty} (-1)^k t^{2k}$$

After integrating this equality over the interval $[0,x]$ with $x \in (-1,1)$ we get

$$\arctan x = \int_{[0,x]} \frac{dt}{1+t^2} = \int_{[0,x]} \sum_{k=0}^{\infty} (-1)^k t^{2k} dt = \sum_{k=0}^{\infty} (-1)^k \int_{[0,x]} t^{2k} dt = \sum_{k=0}^{\infty} \frac{x^{2k+1}}{2k+1}$$

▷

Problem 1.3.4 *Compute $d((xy)^{xy})$.*

◁ Denote $f(x,y) = (xy)^{xy}$, then $\ln f(x,y) = xy \ln(xy)$. After taking differentials we get

$$\frac{df(x,y)}{f(x,y)} = d(xy)\ln(xy) + xy \cdot d(\ln(xy)) = (xdy + ydx)\ln(xy) + xy\frac{xdy + ydx}{xy}$$

Hence

$$df(x,y) = (xy)^{xy}(\ln(xy) + 1)(xdy + ydx)$$

▷

Problem 1.3.5 *Compute* $\int \arctan(x)dx$

◁ We use integration by parts

$$\int \arctan(x)dx = x\arctan(x) - \int xd(\arctan(x)) = x\arctan(x) - \int \frac{x}{1+x^2}dx =$$

$$= x\arctan(x) - \frac{1}{2}\int \frac{d(1+x^2)}{1+x^2} = x\arctan(x) - \frac{1}{2}\ln(1+x^2)$$

▷

Problem 1.3.6 *Compute* $\int_{\mathbb{R}} e^{-x^2}dx$

◁ Denote $I = \int_{\mathbb{R}} e^{-x^2}dx$, then

$$I^2 = \left(\int_{\mathbb{R}} e^{-x^2}dx\right)\left(\int_{\mathbb{R}} e^{-y^2}dy\right) = \int_{\mathbb{R}} e^{-x^2-y^2}dxdy$$

Now we use polar coordinates

$$I^2 = \int_{\mathbb{R}_+\times[0,2\pi]} e^{-r^2}rdrd\varphi = \int_{[0,2\pi]} d\varphi \int_{\mathbb{R}_+} re^{-r^2}dr = 2\pi\frac{1}{2}\int_{\mathbb{R}_+} e^{-r^2}d(r^2) = \pi e^{-r^2}\Big|_0^{+\infty} = \pi$$

Hence $I = \sqrt{\pi}$ ▷

1.4 Combinatorics

Problem 1.4.1 *How many matrices of size $M \times N$ are there with entries equal to ± 1 such that the product of elements in each row and each column is 1*

◁ Each such matrix A can be written as $[(-1)^{P_{i,j}}]$ for some matrix $[P_{i,j}]$ with entries in $\mathbb{Z}/2\mathbb{Z}$ and the property that

$$\sum_{i=1}^{N} P_{i,j} \equiv 0 \pmod 2 \qquad \sum_{j=1}^{M} P_{i,j} \equiv 0 \pmod 2$$

Clearly this correspondence between A and P is bijective, so it is enough to count matrices P. We can arbitrarily choose submatrix of P consisting of first $N-1$ rows and first $M-1$ columns. In this case elements $(P_{i,M})_{i\in\mathbb{N}_{N-1}}$ and $(P_{N,j})_{j\in\mathbb{N}_{M-1}}$ are uniquely determined by equalities given above, i.e.

$$P_{i,M} \equiv -\sum_{j=1}^{M-1} P_{i,j} \pmod 2 \qquad P_{N,j} \equiv -\sum_{i=1}^{N-1} P_{i,j} \pmod 2$$

Constraints put on matrix P allows us to determine $P_{N,M}$ via two equalities

$$P_{N,M} \equiv -\sum_{i=1}^{N-1} P_{i,M} \pmod 2 \qquad P_{N,M} \equiv -\sum_{j=1}^{M-1} P_{N,j} \pmod 2$$

We need to check that both formulae give the same value for $P_{N,M}$. They are indeed equal

$$-\sum_{i=1}^{N-1} P_{i,M} \equiv -\sum_{i=1}^{N-1}\left(-\sum_{j=1}^{M-1} P_{i,j}\right) \equiv \sum_{i=1}^{N-1}\sum_{j=1}^{M-1} P_{i,j}$$

$$\equiv \sum_{j=1}^{M-1}\sum_{i=1}^{N-1} P_{i,j} \equiv -\sum_{j=1}^{M-1}\left(-\sum_{i=1}^{N-1} P_{i,j}\right) \equiv -\sum_{j=1}^{M-1} P_{N,j}$$

Therefore matrix P is completely determined by its submatrix consisting of the first $N-1$ rows and $M-1$ columns. What is more values of its entries can be arbitrary. There are $2^{(N-1)\cdot(M-1)}$ such matrices, so this is the number of all possible matrices P and all possible matrices A. ▷

Problem 1.4.2 *Compute $\sum_{i=0}^{k} i^r$ for $r \in \mathbb{N}^*$.*

◁ We will derive the recurrence formula. For the beginning

$$S_0(k) = \sum_{i=0}^{k} i^0 = k+1$$

Now to derive the recurrence equation note that

$$(i+1)^{r+1} - i^{r+1} = \left(\sum_{p=0}^{r+1}\binom{r+1}{p} 1^{r+1-p} i^p\right) - i^{r+1} = \left(\sum_{p=0}^{r+1}\binom{r+1}{p} i^p\right) - i^{r+1} = \sum_{p=0}^{r}\binom{r+1}{p} i^p$$

Now lets sum this equalities by i from 0 to k. Then we get

$$\sum_{i=0}^{k}\left((i+1)^{r+1}-i^{r+1}\right)=\sum_{i=0}^{k}\sum_{p=0}^{r}\binom{r+1}{p}i^p=\sum_{p=0}^{r}\binom{r+1}{p}\sum_{i=0}^{k}i^p=\sum_{p=0}^{r}\binom{r+1}{p}S_p(k)=$$

$$\binom{r+1}{r}S_r(k)+\sum_{p=0}^{r-1}\binom{r+1}{p}S_p(k)=(r+1)S_r(k)+\sum_{p=0}^{r-1}\binom{r+1}{p}S_p(k)$$

Note that $\sum\limits_{i=0}^{k}\left((i+1)^{r+1}-i^{r+1}\right)$ is a telescopic sum and it equals to $(k+1)^{r+1}$. So we get the following equality

$$(k+1)^{r+1}=(r+1)S_r(k)+\sum_{p=0}^{r-1}\binom{r+1}{p}S_p(k)$$

This gives us

$$S_r(k)=\frac{1}{r+1}\left((k+1)^{r+1}-\sum_{p=0}^{r-1}\binom{r+1}{p}S_p(k)\right)$$

▷

1.5 Probability theory

Problem 1.5.1 *Let p and q be probabilities of heads and tails respectively in a coin trials. Find the expected value of heads in n tosses.*

◁ Let X_i be a random variable defined by equality

$$X_i = \begin{cases} 1 & \text{there was head in } i\text{-th toss} \\ 0 & \text{otherwise} \end{cases}$$

Obviously,

$$\mathbb{E}[X_i] = 1 \cdot \mathbb{P}(X_i = 1) + 0 \cdot \mathbb{P}(X_i = 0) = p$$

Let X be a random variable equal to the number of heads after n tosses, then

$$X = \sum_{i=1}^{n} X_i$$

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] = np$$

▷

Problem 1.5.2 *There are 20 cards in a deck with distinct numbers from 0 to 19 on them. Five cards where taken. Find the expected value of the sum of numbers on these cards.*

◁ Let X_i where $i \in \mathbb{N}_5$ be random variables equal to the number on the i-th card. Obviously,

$$\mathbb{P}(X_i = k) = 0.05 \qquad k \in \mathbb{N}_{19}^*$$

$$\mathbb{E}[X_i] = \sum_{k=0}^{19} k\mathbb{P}(X_i = k) = 0.05 \sum_{k=0}^{19} k = 9.5$$

Let X be a random variable equal to the sum of numbers on five cards taken, then

$$X = \sum_{i=1}^{5} X_i$$

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{5} X_i\right] = \sum_{i=1}^{5} \mathbb{E}[X_i] = 5 \cdot 9.5 = 47.5$$

▷

Problem 1.5.3 *A drunk guy moves along the line. Each moment he makes moves to the right or to the left with probabilities p and $1 - p$ respectively. N steps to the left from the guy there is a precipice. Find the probability of drunk guy falling into the abyss.*

◁ Let $P_k(s)$ be the probability of a drunk guy falling into the abyss after making at most k moves provided his current position is s steps from the precipice. Then we have

$$P_k(s) = pP_{k-1}(s-1) + (1-p)P_{k-1}(s+1)$$

with initial conditions $P_0(s) = 0$ for $s < N$ and $P_k(N) = 1$ for all $k \in \mathbb{N}$. Since the number of possible moves is not bounded, we need to consider asymptotical behaviour of the probability, i.e. find the closed form of $P(s) = \lim_{k\to\infty} P_k(s)$. So we have the following recurrence equation

$$P(s) = pP(s-1) + (1-p)P(s+1)$$

with bounddary condition $P(N) = 1$. This is a recurrence equation of the second order with characteristic equation

$$(1-p)\lambda - 1 + \frac{p}{\lambda} = 0$$

Its roots are $\lambda = p/(1-p)$ and $\lambda = 1$. So

$$P(s) = C_1\left(\frac{p}{1-p}\right)^s + C_2$$

Since $P(N) = 1$ we get $C_2 = 1 - C_1(p/(1-p))^N$, so

$$P(s) = 1 + C_1\left(\left(\frac{p}{1-p}\right)^s - \left(\frac{p}{1-p}\right)^N\right)$$

Clearly, $\lim\limits_{s\to\infty} P(s) = 0$. If $p \geq 0.5$, this is possible iff $C_1 = 0$, hence $P(s) = 1$. If $p < 0.5$, then we get an equation

$$1 - C_1\left(\frac{p}{1-p}\right)^N = 0$$

from which we get $C_1 = \left(\frac{1-p}{p}\right)^N$ and $P(s) = \left(\frac{p}{p-1}\right)^{s-N}$. Finally we get

$$P(s) = \begin{cases} 1 & \text{if } 0.5 \leq p \leq 1 \\ \left(\frac{p}{p-1}\right)^{s-N} & \text{if } 0 \leq p < 0.5 \end{cases}$$

The desired probability equals

$$P(0) = \begin{cases} 1 & \text{if } 0.5 \leq p \leq 1 \\ \left(\frac{1-p}{p}\right)^{N} & \text{if } 0 \leq p < 0.5 \end{cases}$$

▷

Problem 1.5.4 *Random variables $X_1, \ldots, X_n$ are independent and uniformly distributed on $[0, 1]$. Find the expected value and the variance of the random variables*

$$\min(X_1, \ldots, X_n) \quad \textit{and} \quad \max(X_1, \ldots, X_n).$$

◁ Denote $X = \max(X_1, \ldots, X_n)$. By f and F we denote the probability density function and the cumulative distributive function of the uniform distribution on $[0, 1]$. So

$$f(x) = \begin{cases} 1 & \text{if } x \in [0,1] \\ 0 & \text{if } x \notin [0,1] \end{cases} \qquad F(x) = \begin{cases} 1 & \text{if } x < 0 \\ x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x > 1 \end{cases}$$

Now we find probability density function of X. For all $x \in \mathbb{R}$ we have

$$F_X(x) = \mathbb{P}(\{X < x\}) = \mathbb{P}(\{\max(X_1, \ldots, X_n) < x\}) = \mathbb{P}(\{X_1 < x, \ldots, X_n < x\}) =$$

$$= \mathbb{P}(\{X_1 < x\}) \cdot \ldots \cdot \mathbb{P}(\{X_n < x\}) = F_{X_1}(x) \cdot \ldots \cdot F_{X_n}(x) = F^n(x)$$

$$f_X(x) = (F_X(x))' = nF^{n-1}(x)F'(x) = nF^{n-1}(x)f(x)$$

Now we compute the expected value and the variance of X

$$\mathbb{E}[X] = \int_{\mathbb{R}} x f_X(x)dx = \int_{\mathbb{R}} xnF^{n-1}(x)f(x)dx = \int_{[0,1]} xnx^{n-1}dx = \frac{n}{n+1}$$

$$\mathbb{E}[X^2] = \int_{\mathbb{R}} x^2 f_X(x)dx = \int_{\mathbb{R}} x^2 nF^{n-1}(x)f(x)dx = \int_{[0,1]} x^2 nx^{n-1}dx = \frac{n}{n+2}$$

$$\mathbb{V}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \frac{n}{n+1} - \left(\frac{n}{n+2}\right)^2$$

Denote $Y = \min(X_1, \ldots, X_n)$ and we shall find cummulative and the probability density function of Y. For all $x \in \mathbb{R}$ we have

$$F_Y(x) = \mathbb{P}(\{Y < x\}) = 1 - \mathbb{P}(\{Y \geq x\}) = 1 - \mathbb{P}(\{\min(X_1, \ldots, X_n) \geq x\}) =$$

$$= 1 - \mathbb{P}(\{X_1 \geq x, \ldots, X_n \geq x\}) = 1 - \mathbb{P}(\{X_1 \geq x\}) \cdot \ldots \cdot \mathbb{P}(\{X_n \geq x\})$$

$$= 1-(1-\mathbb{P}(\{X_1 < x\}))\cdot\ldots\cdot(1-\mathbb{P}(\{X_n < x\})) = 1-(1-F_{X_1}(x))\cdot\ldots\cdot(1-F_{X_n}(x)) = 1-(1-F(x))^n$$

$$f_Y(x) = (F_Y(x))' = (1-(1-F(x))^n)' = -n(1-F(x))^{n-1}(-F'(x)) = n(1-F(x))^{n-1}f(x)$$

Now we compute the expected value and the variance of Y

$$\mathbb{E}[Y] = \int_{\mathbb{R}} x f_Y(x)dx = \int_{\mathbb{R}} xn(1-F(x))^{n-1}dx = \int_{[0,1]} xn(1-x)^{n-1}dx = \int_{[0,1]} (1-t)nt^{n-1}dt$$

$$= 1 - \frac{n}{n+1} = \frac{1}{n+1}$$

$$\mathbb{E}[Y^2] = \int_{\mathbb{R}} x^2 f_Y(x)dx = \int_{\mathbb{R}} x^2 n(1-F(x))^{n-1}dx = \int_{[0,1]} x^2 n(1-x)^{n-1}dx = \int_{[0,1]} (1-t)^2 nt^{n-1}dt$$

$$= 1 - \frac{2n}{n+1} + \frac{n}{n+2} = \frac{2}{(n+1)(n+2)}$$

$$\mathbb{V}[Y] = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 = \frac{1}{n+1} - \left(\frac{2}{(n+1)(n+2)}\right)^2$$

▷

Problem 1.5.5 *There are three towers in apexes of the equilateral triangle. The dragon sits on the first tower. With equal probabilities dragon flies to two other towers. Find the expected value of the number flights after which the dragon will come back to the first tower.*

◁ Clearly in order to come back to the first tower the dragon must make at least two flights. Assume he made n flights. There are two scenarios: on the first flight the dragon flies to the second tower, or the dragon flies to the third tower. Both scenarios have the same probability 0.5^{n-2}, so the total probability is 0.5^{n-1}. The expected value of flights is

$$\sum_{n=2}^{\infty} n0.5^{n-1} = \sum_{n=2}^{\infty} nx^{n-1}\bigg|_{x=0.5} = \left(\frac{d}{dx}\sum_{n=2}^{\infty} x^n\right)\bigg|_{x=0.5} = \frac{d}{dx}\left(\frac{x^2}{1-x}\right)\bigg|_{x=0.5} = \frac{2x-x^2}{(1-x)^2}\bigg|_{x=0.5} = 3$$

▷

Problem 1.5.6 *There are n places on the airplane. An old crazy lady comes first and randomly chooses a sit. Then come the other passengers. If a passenger sees his place is already taken then he arbitrarily chooses another sit. Find the probability that the last passenger will sit at his place.*

◁ Let $p(n)$ be the probability that the last passenger will not sit at his place in case of n-man problem. An lady woman can choose any sit with probability $1/n$. If she chooses her own sit then $p(n) = 1$. If she chooses a sit of the last passenger, then $p(n) = 0$. If she chooses k-th place where $k \in \{2, \ldots, n-1\}$, then all the passengers with numbers $2, \ldots, k-1$ will sit at their places. The k-th passenger will choose its place randomly. Hence we reduce our problem to the $n-k$-man problem, where the role of an old woman is played by k-th passenger. Thus the total probability is

$$p(n) = 1 \cdot \frac{1}{n} + p(n-1) \cdot \frac{1}{n} + \ldots + p(1) \cdot \frac{1}{n} = \frac{1}{n}\left(1 + \sum_{k=2}^{n-1} p(k)\right)$$

Now we prove by induction that $p(n) = 1/2$. Obviously, $p(2) = 1/2$. Assume that $p(k) = 1/2$ for $k \in \mathbb{N}_m$, then

$$p(m+1) = \frac{1}{m+1}\left(1 + \sum_{k=2}^{m} p(k)\right) = \frac{1}{m+1}\left(1 + \sum_{k=2}^{m} \frac{1}{2}\right) = \frac{1}{2}$$

The desired probability is $1 - p(n) = 1 - 1/2 = 1/2$. ▷

1.6 Programming

Problem 1.6.1 *Write a program which computes the greatest common divisor of two integers. Prove correctness.*

◁ This algorithm is called the Euclidean algorithm.

$$\gcd(a,b) = \begin{cases} a & \text{if } b = 0 \\ \gcd(b, a \mod b) & \text{if } b \neq 0 \end{cases}$$

Since each iteration of the algorithm decreases the second argument, then this algorithm will always stop. To prove correctness we need to show that $\gcd(a,b) = \gcd(b, a \mod b)$. It enough to show that the sets of common divisors of a, b and b, $a \mod b$ coincide. Assume $d|a$ and $d|b$. Since $a \mod b = a - kb$ for some $k \in \mathbb{Z}$, then $d|(a \mod b)$. Thus $d|b$ and $d|(a \mod b)$. Now assume $d|b$ and $d|(a \mod b)$. Since $a = kb + (a \mod b)$ for some $k \in \mathbb{Z}$, then $d|a$. Thus $d|a$ and $d|b$.

```
#include <iostream>

int gcd(int a, int b)
{
    while(b !=0 )
    {
        a %= b;
        std::swap(a,b);
    }
    return a;
}

int main()
{
    int a, b;
    std::cin >> a >> b;
    std::cout << gcd(a, b);

    return 0;
}
```

▷

Problem 1.6.2 *Write a program which finds inverse element in a finite field.*

◁ Assume we are given a finite field $\mathbb{F}_k$, then by little Fermat's theorem for all $x \in \mathbb{F}_k$ we have

$$x^{k-1} = e$$

Hence $x^{-1} = x^{p-2}$. It is remains to write a fast powerization in finite fields.

```
#include <iostream>
#include <vector>
int power_mod(int x, int degree, int k)
{
    if (degree == 0)
    {
        return 1;
    }
    if (degree % 2 == 1)
    {
        return (x * power_mod(x, degree - 1, k)) % k;
    }
    int half_power = power_mod(x, degree / 2, k);
    return (x * x) % k;
}

int inverse_mod(int x, int k)
{
    return power_mod(x, k - 2, k);
}

int main()
{
    int x, k;
    std::cin >> x >> k;
    std::cout << inverse_mod(x, k);
    return 0;
}
```

▷

Problem 1.6.3 *Let a be an array of integers of length n. Write a program which effectively finds the sum of elements in each of m consecutive requests of the form (begin,end).*

◁ The main idea is to store additional array of sums of elements of each prefix of array.

```
#include <iostream>
#include <vector>

class accumulator
{
public:
    accumulator(const std::vector<int>& numbers)
      : numbers_(numbers), sums_(numbers.size() + 1, 0)
    {}

    long long get_sum(size_t begin, size_t end)
```

```
    {
        return end >= begin ? sums_[end + 1] - sums_[begin] : 0;
    }

    void initialize()
    {
        for (size_t i = 1; i < numbers_.size(); ++i)
        {
            sums_[i] = sums_[i - 1] + numbers_[i - 1];
        }
    }

private:
    const std::vector<int>& numbers;
    std::vector<long long> sums_;
};

int main()
{
    size_t n;
    std::cin >> n;
    std::vector<int> numbers(n, 0);
    for (size_t i = 0; i < numbers.size(); ++i)
    {
        std::cin >> numbers[i];
    }

    accumulator accumulator_engine(numbers);
    accumulator_engine.initialize();

    size_t m;
    std::cin >> m;
    for (size_t i = 0; i < m; ++i)
    {
        int begin, end;
        std::cin >> begin >> end;
        std::cout << accumulator_engine.get_sum(begin, end);
    }

    return 0;

}
```

▷

Problem 1.6.4 *There are $2n + 1$ integers in the array, with one unique integer and n integers that repeat twice. Write a program with $O(n)$ complexity which finds this integer with usage of*

$O(1)$ *memory.*

◁ Consider operation $\oplus$ which makes bitwise addition modulo 2 of binary representations of integers. This is a commutative associative operation. Note that for all $x \in \mathbb{N}^*$ holds

$$x \oplus x = 0 \qquad x \oplus 0 = x$$

Let $(a_i)_{i \in \mathbb{N}_n}$ be elements of array. Let $\sigma \in S_{2n+1}$ be such permutation of elements of array that after permutation twin integers will follow one by one and the unique integer will occur at the end. Then

$$\bigoplus_{i=1}^{2n+1} a_i = \bigoplus_{i=1}^{2n+1} a_{\sigma(i)} = \left(\bigoplus_{k=1}^{n} (a_{\sigma(2k-1)} \oplus a_{\sigma(2k)}) \right) \oplus a_{\sigma(2n+1)} = \left(\bigoplus_{k=1}^{n} 0 \right) \oplus a_{\sigma(2n+1)} = a_{\sigma(2n+1)}$$

Thus the unique element of array is equal to

$$\bigoplus_{i=1}^{2n+1} a_i$$

```
#include <iostream>

int main()
{
    int n;
    std::cin >> n;
    int unique_element = 0;
    for (size_t i = 0; i < 2 * n + 1; ++i)
    {
        int integer;
        std::cin >> integer;
        unique_element ^= integer;
    }
    std::cout << unique_element;
    return 0;
}
```

▷

Problem 1.6.5 *Write a program with $O(\log n)$ complexity, which computes n-th power of a given integer.*

◁ In order to powerize quickly we will use the following identities

$$a^{2k+1} = aa^{2k} \qquad a^{2k} = (a^k)^2$$

```
#include <iostream>
#include <vector>

long long power(int integer, int degree)
```

```
{
    if (degree == 0)
    {
        return 1;
    }
    if (degree % 2 == 1)
    {
        return integer * power(integer, degree - 1);
    }

    long long half_power = power(integer, degree / 2);
    return half_power * half_power;
}

int main()
{
    int integer, degree;
    std::cin >> integer >> degree;
    std::cout << power(integer, degree);
    return 0;
}
```

▷

Problem 1.6.6 *Write a program with $O(\log n)$ complexity which finds n-th Fibbonacci number*

◁ Let's rewrite the recurrence relation for Fibbonacci numbers $f_n = f_{n-1} + f_{n-2}$ in the matrix form.

$$\begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} f_{n-1} \\ f_{n-2} \end{pmatrix}$$

Then we get

$$\begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} f_1 \\ f_0 \end{pmatrix}$$

It remains to quickly compute $n-1$-th power of the matrix

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

which is easily done with the help of identities

$$a^{2k} = aa^{2k} \qquad a^{2k} = \left(a^k\right)^2$$

```
#include <iostream>
#include <vector>

class matrix2x2
{
```

```
public:

    matrix2x2() : matr_(2,std::vector<int>(2,0))
    {}

    matrix2x2(int matr00, int matr01, int matr10, int matr11)
      : matr_(2, std::vector<int>(2,0))
    {
        matr_[0][0] = matr00;
        matr_[0][1] = matr01;
        matr_[1][0] = matr10;
        matr_[1][1] = matr11;
    }

    matrix2x2(const matrix2x2& matrix)
    {
        matr_ = matrix.matr_;
    }

    const matrix2x2& operator = (const matrix2x2& rhs)
    {
        if (this != &rhs)
        {
            matr = rhs.matr;
        }
        return *this;
    }

    int& operator() (size_t i, size_t j)
    {
        return matr_[i][j];
    }

    const int& operator()(size_t i, size_t j) const
    {
        return matr_[i][j];
    }

    matrix2x2& operator * (const matrix2x2& ths)
    {
        return product(*this, rhs);;
    }

private:

    matrix2x2 product(const matrix2x2& lhs, const matrix2x2& rhs) const
    {
```

```
        retrn matrix2x2(lhs(0, 0) * rhs(0, 0) + lhs(0, 1) * rhs(1, 0),
                        lhs(0, 0) * rhs(0, 1) + lhs(0, 1) * rhs(1, 1)
                        lhs(1, 0) * rhs(0, 0) + lhs(1, 1) * rhs(1, 0)
                        lhs(1, 0) * rhs(0, 1) + lhs(1, 1) * rhs(1, 1));
    }


private:
    std::vector<int> matr_;
};

matrix2x2 matrix_power(const matrix2x2& matrix, size_t power)
{
 if (power == 0)
    {
     return matrix2x2(1, 0, 0, 1);
    }
    if (power % 2 == 1)
    {
     return matrix_power(matrix, power - 1) * matrix;
    }

    matrix2x2 half_power = matrix_power(matrix, power / 2);
    return half_power * half_power;
}

long long fibbonacci_number(int n)
{
    matrix2x2 fibbonacci_matrix = matrix_power(matrix2x2(1, 1, 1, 0), n - 1);
    return fibbonacci_matrix(0,0);
}

int main()
{
    size_t order;
    std::cin >> order;
    std::cout << fibbonacci_number(order);
    return 0;
}
```

▷

Problem 1.6.7 *There is an array of n integers. It is known that some number repeats in the array more than $n/2$ times. Give an algorithm that finds this number. Time complexity $O(n)$, memory restrictions $O(1)$.*

◁ We call popular the number we are looking for. First we present the algorithm and then explain why it is correct.

```
#include <iostream>
#include <vector>

int find_popular_number(const std::vector<int>& values)
{
    int popular_number;
    size_t counter = 0;

    for (size_t number_index = 0; number_index < values.size(); ++number_index)
    {
        if (counter == 0)
        {
            popular_number = values[number_index];
            counter++;
        }
        else
        {
            counter += popular_number != values[number_index] ? -1 : 1;
        }
    }

    return popular_number;
}

int main()
{
    size_t values_count;
    std::cin >> values_count;
    std::vector<int> values(values_count);
    for (size_t index = 0; index < values.size(); ++index)
    {
        std::cin >> values[index];
    }

    std::cout << find_popular_number(values);

    return 0;
}
```

If the counter is equal to 1 we are looking for the popular number. And we set the popular number to be the first incoming number. If meet the number equal to the current popular we increase the counter, otherwise we decrease it. Each decrease of the counter is equivalent to throwing away the number at current iteration and the current popular number. By assumption we have at least $n/2$ equal numbers, hence such reduction will exclude all non popular numbers, and the remaining current popular number will be the genuine popular number. ▷

2 Entrance exams in SDA

2.1 Exam on 10.06.2012

Problem 2.1.1 *There are 2012 weights with different masses. They are divided into two groups with 1006 weights in each group. Weights in each group are sorted in ascending order. Give an algorithm that finds 1006's weight among all weights with at most 11 weighings.*

◁ Denote that groups A and B. Without loss of generality we may assume that $A[503] > B[503]$ (this is one weighing). Clearly the desired weight can not belong to $B[1 \dots 502]$ or $A[504 \dots 1006]$. So we reduce the problem to finding that element in $A[1 \dots 503]$ and $B[503 \dots 1006]$ with 10 weighings left. We can repeat the procedure until we arrive at comparison of two single element arrays. Since $2^{11} > 2012$ we will definitely reach this comparison. ▷

Problem 2.1.2 *Compute* $\int_0^{2\pi} \sin^8 x dx$.

◁

$$\int_0^{2\pi} \sin^8 x dx = \int_0^{2\pi} \left(\frac{e^{ix} - e^{-ix}}{2i}\right)^8 dx$$

$$= \int_0^{2\pi} 2^{-8}(e^{ix} - e^{-ix})^8 dx = 2^{-8} \int_0^{2\pi} \sum_{k=0}^{8} \binom{8}{k} (e^{ix})^k (-e^{-ix})^{8-k} dx$$

$$= 2^{-8} \sum_{k=0}^{8} \binom{8}{k} (-1)^{8-k} \int_0^{2\pi} e^{i(2k-8)x} dx = 2^{-8} \sum_{k=0}^{8} \binom{8}{k} (-1)^k 2\pi \delta_{2k-8,0} = 2^{-8} \binom{8}{4} (-1)^4 2\pi = \frac{\binom{8}{4}}{2^7}\pi$$

▷

Problem 2.1.3 *Assume a polynomial* $f \in \mathbb{R}[x]$ *attains only positive values. Prove it can be represented as a sum of squares of some polynomials.*

◁ By fundamental theorem of algebra $f(x) = A\prod_{i=1}^{n}(x - a_i)\prod_{j=1}^{m}(x^2 + p_j x + q_j)$ for some real numbers A, $(a_i)_{i\in\mathbb{N}_n}$, $(p_j)_{j\in\mathbb{N}_m}$ and $(q_j)_{j\in\mathbb{N}_m}$, where $p_j^2 - 4q_j < 0$ for all $j \in \mathbb{N}_m$. Clearly, $f(a_i) = 0$ for all $i \in \mathbb{N}_n$, which contradicts assumption of positivity, hence $n = 0$ and therefore $f(x) = A\prod_{j=1}^{m}(x^2 + p_j x + q_j)$. Since $p_j^2 - 4q_j < 0$ for all $j \in \mathbb{N}_m$, then for all $x \in \mathbb{R}$ and all $j \in \mathbb{N}_m$ we have $x^2 + p_j x + q_j > 0$. Again, since f is always positive we conclude that $A > 0$. Without loss of generality we may assume that $A = 1$.

Now we prove by induction on m that each polynomial of the form $f(x) = \prod_{j=1}^{m}(x^2 + p_j x + q_j)$ can be represented as a sum of two squared polynomials. Basis of induction: $m = 1$. We have

$$f(x) = x^2 + px + q = \left(x + \frac{p}{2}\right)^2 + \left(\frac{\sqrt{4q - p^2}}{2}\right)^2$$

and we are done. Assume our claim is true for $m > 1$. Consider polynomial $f(x) = \prod_{j=1}^{m+1}(x^2 + p_j x + q_j)$. By inductive assumption there exist polynomials g_1, g_2 such that

$$\prod_{j=1}^{m}(x^2 + p_j x + q_j) = g_1^2(x) + g_2^2(x)$$

Denote $h_1(x) = x + \frac{1}{2}p_{m+1}$, $h_2(x) = \frac{1}{2}\sqrt{4q_{m+1} - p_{m+1}^2}$, then

$$f(x) = \left(\prod_{j=1}^{m}(x^2 + p_j x + q_j)\right)(x^2 + p_{m+1} + q_m) = (g_1^2(x) + g_2^2(x))(h_1^2(x) + h_2^2(x))$$

$$= (g_1(x)h_1(x) + g_2(x)h_2(x))^2 + (g_1(x)h_2(x) - g_2(x)h_1(x))^2$$

Induction step completed. ▷

Problem 2.1.4 *What is the maximal possible variance of a random variable X with values in $[0, 1]$*

◁ Assume a random variable is defined on probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Denote $m = \mathbb{E}[X]$, then

$$\mathbb{V}[X] - \int_\Omega \left(X(\omega) - \frac{1}{2}\right)^2 d\mathbb{P} = \int_\Omega \left((X(\omega) - m)^2 - \left(X(\omega) - \frac{1}{2}\right)^2\right) d\mathbb{P}$$

$$= \int_\Omega \left(X(\omega)(1 - 2m) + m^2 - \frac{1}{4}\right) d\mathbb{P} = (1 - 2m)\int_\Omega X(\omega)d\mathbb{P} + \left(m^2 - \frac{1}{4}\right)\int_\Omega d\mathbb{P}$$

$$= (1 - 2m)m + m^2 - \frac{1}{4} = -\left(m - \frac{1}{2}\right)^2 \leq 0$$

From these inequalities it follows that $\mathbb{V}[X]$ is maximal if $m = 1/2$. Since X takes values in $[0, 1]$, then $\left(X(\omega) - \frac{1}{2}\right)^2 \leq \frac{1}{4}$. Thus the upper bound for $\mathbb{V}[X]$ is $\frac{1}{4}$. Now assume there is an event $A_1 \in \mathcal{F}$ with $\mathbb{P}(A_1) = \frac{1}{2}$. Consider random variable defined as $X = \chi_{A_1}$, then

$$\mathbb{V}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \int_\Omega \chi_{A_1}(\omega)d\mathbb{P} - \left(\int_\Omega \chi_{A_1}^2(\omega)d\mathbb{P}\right) = \mathbb{P}(A_1) - (\mathbb{P}(A_1))^2 = \frac{1}{4}$$

Thus X is a random variable with values in $[0, 1]$ and the maximal possible variance equals to $\frac{1}{4}$. ▷

Problem 2.1.5 *In the group of n people each man may know or may not know any other. All acquaintances are represented via boolean $n \times n$ matrix. We say that a man is a celebrity if all others know him, but he doesn't know anyone. Give an algorithm that finds a celebrity or says it doesn't exist. Algorithms complexity $O(n)$, memory restriction $O(1)$.*

◁ Notice that in the group there can be at most one celebrity. So we'll find that candidate and check whether he is a real celebrity. Let a be the boolean matrix representing acquaintances. The main idea of the algorithm is the fact that if $a[i][j] = 0$ ($a[i][j] = 1$), then j (i) is not a celebrity.

```
#include <iostream>
#include <vector>

size_t find_celebrity(
    const std::vector<std::vector<bool>>& acquaintances)
{
    size_t crowd_size = acquaintances.size();
```

```
    // find the candidate
    size_t candidate = 0;
    for (size_t man = 1; man < crowd_size; ++man)
    {
        if (acquaintances[candidate][man] == true)
        {
            candidate = man;
        }
    }

    //check candidate
    for (size_t man = 0; man < crowd_size; ++man)
    {
        if (acquaintances[man][candidate] == false ||
           (man != candidate &&
            acquaintances[candidate][man] == true))
        {
            candidate = -1;
            break;
        }
    }

    return candidate;
}

int main()
{
    size_t crowd_size;
    std::cin >> crowd_size;
    std::vector<std::vector<bool>> acquaintances(
        crowd_size, std::vector<bool>(crowd_size, true));

    for (size_t man = 0; man < acquaintances.size(); ++man)
    {
        for (size_t buddy = 0;
            buddy < acquaintances[man].size();
            ++buddy)
        {
            size_t acquaintance;
            std::cin >> acquaintance;
            acquaintances[man][buddy] = (
                acquaintance != 0 ? true : false);
        }
    }

    std::cout << find_celebrity(acquaintances);
    return 0;
```

}

▷

Problem 2.1.6 *Consider arbitrary permutation of n elements. Prove that any k given elements will belong to the same circle with probability $1/k$.*

◁ Let's count number of permutations of n elements that contain cycle of length $k+i$ with k given elements. Denote this quantity $c_n(i)$. There are $\binom{n-k}{i}$ ways to choose i elements that together with given k elements will constitute a cycle. There are $(i+k-1)!$ ways to build a cycle from k given elements and i chosen elements. Finally, there are $(n-k-i)!$ ways to arrange permutation of remaining $n-k-i$ elements. Therefore $c_n(i) = \binom{n-k}{i}(i+k-1)!(n-k-i)!$. The desired probability is

$$p_n(k) = \frac{1}{n!}\sum_{i=0}^{n-k} c_n(i)$$

$$= \frac{1}{n!}\sum_{i=0}^{n-k}\binom{n-k}{i}(i+k-1)!(n-k-i)! = \frac{1}{n!}\sum_{i=0}^{n-k}\frac{(n-k)!}{(n-k-i)!i!}(i+k-1)!(n-k-i)!$$

$$= \frac{(n-k)!}{n!}\sum_{i=0}^{n-k}\frac{(i+k-1)!}{i!} = \frac{(n-k)!(k-1)!}{n!}\sum_{i=0}^{n-k}\frac{(i+k-1)!}{i!(k-1)!} = \frac{(n-k)!(k-1)!}{n!}\sum_{i=0}^{n-k}\binom{i+k-1}{i}$$

Note that

$$\sum_{i=0}^{p}\binom{i+q}{i} = \binom{q}{0} + \sum_{i=1}^{p}\binom{i+q}{i} = \binom{q+1}{0} + \sum_{i=1}^{p}\left(\binom{i+1+q}{i} - \binom{i+q}{i-1}\right)$$

$$= \binom{q+1}{0} + \binom{p+1+q}{p} - \binom{q+1}{0} = \binom{p+1+q}{p}$$

so

$$p_n(k) = \frac{(n-k)!(k-1)!}{n!}\sum_{i=0}^{n-k}\binom{i+k-1}{i} = \frac{(n-k)!(k-1)!}{n!}\binom{n}{n-k}$$

$$= \frac{(n-k)!(k-1)!}{n!}\frac{n!}{(n-k)!k!} = \frac{1}{k}$$

▷

Problem 2.1.7 *There is an unknown quadratic form Q on $\mathbb{R}^n$. One may ask questions of the form 'What is the value of $Q(v)$?'. What is the minimal number of questions required to determine if Q is positive definite?*

◁ For the beginning we claim that one needs to know all the entries of the matrix of quadratic form to say if it is positive definite. Indeed, if at least one entry is not fixed, then we can fix all the remaining entries and consider sufficiently big positive values of non fixed entry, and sufficiently big negative values of non fixed entry. For these values of non fixed entry quadratic form will have different definiteness type. Therefore to give a definite answer whether quadratic form Q is positive definite we need to know all its entries. Since the matrix q of the quadratic form is symmetric we

need to know only entries over the main diagonal and on the main diagonal. That is $n(n+1)/2$ elements. For the beginning we determine entries of main diagonal by formula the $q_{i,i} = Q(e_i)$, where $i \in \mathbb{N}_n$. This requires n questions. In order to determine entries over the main diagonal we use the formula $q_{i,j} = \frac{1}{2}(Q(e_i + e_j) - Q(e_i) - Q(e_j))$. This will require $\frac{1}{2}(n-1)n$ questions. The total number of questions is $n + \frac{1}{2}n(n-1) = \frac{1}{2}n(n+1)$. We claim this is the minimal number of questions. Indeed, assume there exists an algorithm A that completely determines matrix q with $m < \frac{1}{2}n(n+1)$ questions. Then it uses values of Q on m vectors $v^{(1)}, \ldots, v^{(m)}$. Which gives us m equations of the form $\sum_{i,j=1}^{n} v_i^{(k)} q_{i,j} v_j^{(k)} = Q(v^{(k)})$ where $k \in \mathbb{N}_m$. This is a system of m equations with $\frac{1}{2}n(n+1)$ unknowns which is greater than m. Then there infinitely many solutions of this system of linear equations, i.e. there are infinitely many quadratic forms with values $(Q(v^{(k)}))_{k \in \mathbb{N}_m}$ on vectors $(v^{(k)})_{k \in \mathbb{N}_m}$. Therefore the algorithm A cannot definitely determine the matrix q. Contradiction, hence at least $\frac{1}{2}n(n+1)$ questions required to determine q. As we showed above there is an algorithm that answers the question using exactly $\frac{1}{2}n(n+1)$ questions. $\triangleright$

2.2 Exam on 26.05.2013

Problem 2.2.1 *Find the following recurrent sequence*

$$x_0 = 0, \quad x_1 = 1, \quad x_{n+1} = \frac{x_n + nx_{n-1}}{n+1}$$

is convergent and find its limit.

◁ Consider sequence $y_n = x_{n+1} - x_n$, then $y_n = -\frac{n}{n+1} y_{n-1}$ and $y_0 = 1$. Clearly,

$$y_n = y_0 \cdot \prod_{k=1}^{n} \frac{-k}{k+1} = \frac{(-1)^n}{n+1}$$

hence

$$x_n - x_0 = \sum_{k=0}^{n-1} y_k = \sum_{k=0}^{n-1} \frac{(-1)^k}{k+1}$$

Now we compute the desired limit

$$\lim_{n\to\infty} x_n = \lim_{n\to\infty} \left(x_0 + \sum_{k=0}^{n-1} \frac{(-1)^k}{k+1} \right) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k+1} = \sum_{k=0}^{\infty} (-1)^k \int_0^1 x^k dx = \int_0^1 \sum_{k=0}^{\infty} (-1)^k x^k dx$$

$$= \int_0^1 \sum_{k=0}^{\infty} (-x)^k dx = \int_0^1 \frac{1}{1+x} dx = \ln 2$$

▷

Problem 2.2.2 *Assume we are given* 100 *subsets of* $\{0, 1, \ldots, 9\}$. *Prove there two subsets with symmetric difference not greater than* 2.

◁ Each subset of $\{0, 1, \ldots, 9\}$ can be identified with some vector in $\{0, 1\}^{10}$. Hence the original problem can be reformulated as follows: does there exist 100 vectors which differs from each other in at least 2 positions, i.e. the Hamming distance is greater than 2. We prove that such vectors do not exist. It will be enough to show that there is no 100 disjoint balls of radius 1 in $\{0, 1\}^{10}$.

We let $A_q(n, d)$ denote the maximum number of q-ary vectors of dimension n with minimum pairwise distance d. For a given vector $c \in \{0, \ldots, q\}^n$ we can count q-ary vectors of length n that differs from c in at most t positions. There are $\binom{n}{k}$ ways to choose $0 \le k \le t$ positions where a vector can differ from c. At each position we can place $q-1$ values. So there are $\binom{n}{k}(q-1)^k$ vectors that differ from a given vector c in k positions. Thus there are $\sum_{k=0}^{t} \binom{n}{k}(q-1)^k$ that differ from c in at most t positions. Since we are looking for vectors that differ in at most d positions we need to set $t = (d-1)/2$. Denote $m = \sum_{k=0}^{(d-1)/2} \binom{n}{k}(q-1)^k$. The total number of vectors is q^n, so

$$A_q(n, d) \le \frac{q^n}{m} = \frac{q^n}{\sum_{k=0}^{(d-1)/2} \binom{n}{k}(q-1)^k}$$

Using this bound we get $A_2(10, 3) \le \frac{2^{10}}{1+10} < 94 < 100$, therefore such collection of subsets doesn't exist. ▷

Problem 2.2.3 *A random point P is taken on the unit circle $x^2 + y^2 = 1$. A random point Q is taken from the unit ball $x^2 + y^2 \leq 1$. Let R be a rectangle with diagonal PQ, whose sides are parallel to the coordinates axis. What is the probability that R is contained in the unit ball.*

◁ Let P has coordinates (a, b) and Q has coordinates (c, d), then $a^2 + b^2 = 1$ and $c^2 + d^2 \leq 1$. In order for rectangle to be inside the circle it is necessary and sufficient that $a^2 + d^2 \leq 1$ and $c^2 + b^2 \leq 1$. Therefore $|c| \leq |a|$ and $|d| \leq |b|$. For the fixed point P the point Q must belong to the rectangle $[-|a|, |a|] \times [-|b|, |b|]$. The probablity of this event is

$$\frac{\text{area}(circle)}{\text{area}(rectangle)} = \frac{2|a| \cdot 2|b|}{\pi \cdot 1^2} = \frac{4}{\pi}|ab|$$

Since P is uniformly distributed over the circle, then $a = \cos(2\pi t)$ and $b = \sin(2\pi t)$ with t uniformly disturbed on $[0, 1]$. Therefore the desired probability is

$$p = \int_0^1 \frac{4}{\pi} |\cos(2\pi t)\sin(2\pi t)|dt = \frac{2}{\pi}\int_0^1 |\sin(4\pi t)|dt = \frac{2}{\pi}\int_0^{4\pi} |\sin(z)|\frac{dz}{4\pi} = \frac{1}{2\pi^2}\int_0^{4\pi} |\sin z|dz$$

$$= \frac{4}{2\pi^2}\int_0^{\pi} |\sin z|dz = \frac{2}{\pi^2}\int_0^{\pi} \sin z dz = \frac{4}{\pi^2}$$

▷

Problem 2.2.4 *Let $f : \mathbb{R} \to \mathbb{R}_+$ be a continuous function with the property that $\int_{\mathbb{R}} f(x)dx = 1$. Let $\alpha \in (0, 1)$ and $[a, b]$ is the interval of minimal length such that $\int_{[a,b]} f(x)dx = \alpha$. Show that $f(a) = f(b)$.*

◁ We have a constrained optimization problem. We need to minimize the function $b - a$ with restriction that $\int_{[a,b]} f(x)dx = \alpha$. The Lagrangian of this problem is

$$L(a, b, \lambda) = b - a + \lambda\left(\int_{[a,b]} f(x)dx - \alpha\right)$$

By assumption L attains minimum for some a and b, hence we have $L'_a = L'_b = 0$. This gives us two equations

$$-1 + \lambda(-f(b)) = 0, \qquad 1 + \lambda f(a) = 0$$

If $\lambda = 0$ these equations give a contradiction, so $\lambda \neq 0$. After summation of the equations we obtain $\lambda(f(b) - f(a)) = 0$. Since $\lambda \neq 0$ we have $f(b) = f(a)$. ▷

Problem 2.2.5 *Assume we a given a $n \times n$ matrix M, such that*

$$M_{ij} = \begin{cases} a_i a_j & \text{if} \quad i \neq j \\ a_i^2 + k & \text{if} \quad i = j \end{cases}$$

Find $\det(M)$.

◁ Let X be an $m \times n$ matrix and Y be $n \times m$ matrix, then we have the following identity

$$\begin{pmatrix} kE_n + YX & 0 \\ X & kE_m \end{pmatrix} \begin{pmatrix} E_n & Y \\ 0 & E_m \end{pmatrix} = \begin{pmatrix} E_n & Y \\ 0 & E_m \end{pmatrix} \begin{pmatrix} kE_n & 0 \\ X & kE_m + XY \end{pmatrix}$$

Now we compute the determinant of both sides

$$\det\left(\begin{pmatrix} kE_n + YX & 0 \\ X & kE_m \end{pmatrix} \begin{pmatrix} E_n & Y \\ 0 & E_m \end{pmatrix}\right) = \det\left(\begin{pmatrix} E_n & Y \\ 0 & E_m \end{pmatrix} \begin{pmatrix} kE_n & 0 \\ X & kE_m + XY \end{pmatrix}\right)$$

$$\det\begin{pmatrix} kE_n + YX & 0 \\ X & kE_m \end{pmatrix} \det\begin{pmatrix} E_n & Y \\ 0 & E_m \end{pmatrix} = \det\begin{pmatrix} E_n & Y \\ 0 & E_m \end{pmatrix} \det\begin{pmatrix} kE_n & 0 \\ X & kE_m + XY \end{pmatrix}$$

Recall the following formulae

$$\det\begin{pmatrix} A & B \\ 0 & D \end{pmatrix} = \det(A)\det(D) \qquad \det\begin{pmatrix} A & 0 \\ B & D \end{pmatrix} = \det(A)\det(D)$$

then

$$\det(kE_n + YX)\det(kE_m)\det(E_n)\det(E_m) = \det(E_m)\det(E_n)\det(kE_n)\det(kE_m + XY)$$

$$k^m \det(kE_n + YX) = k^n \det(kE_m + XY)$$

$$\det(kE_n + YX) = k^{n-m} \det(kE_m + XY)$$

Let $X = (a_1, \ldots, a_n)$, $Y = (a_1, \ldots, a_n)^{tr}$, then $M = kE_n + YX$. By previous result we have

$$\det(M) = \det(kE_n + YX) = k^{n-1}\det(kE_1 + XY) = k^{n-1}\left(k + \sum_{i=1}^{n} a_i^2\right)$$

▷

Problem 2.2.6 *Assume we are given a binary $n \times n$ matrix (each entry take 1 bit of memory). We say that a row or column is bad if it contains at least one zero. Give an algorithm which with usage of $O(1)$ memory sets to zero all entries of bad rows and columns.*

◁ Denote that matrix by a. If $a[i][j] = 0$, then i-th row and j-th column are bad. Hence it is enough to iterate through the matrix and each time we met a zero element we set $a[i][0]$ and $a[0][j]$ to zero. After this in the first row and first column we indicate all the bad rows and columns. It remains to set them all to zero.

```
#include <iostream>
#include <vector>

void print_matrix(const std::vector<std::vector<bool>>& matrix)
{
    for (size_t i = 0; i < matrix.size(); ++i)
    {
        for (size_t j = 0; j < matrix[i].size(); ++j)
        {
            std::cout << matrix[i][j] << ' ';
```

```
        }
        std::cout << '\n';
    }
}

void fix_matrix(std::vector<std::vector<bool>>& matrix)
{
    for (size_t i = 0; i < matrix.size(); ++i)
    {
        for (size_t j = 0; j < matrix[i].size(); ++j)
        {
            if (matrix[i][j] == false)
            {
                matrix[0][j] = matrix[i][0] = false;
            }
        }
    }

    for (size_t i = 1; i < matrix.size(); ++i)
    {
        for (size_t j = 1; j < matrix[i].size(); ++j)
        {
            if (matrix[i][0] == false || matrix[0][j] == false)
            {
                matrix[i][j] = false;
            }
        }
    }
}

int main()
{
    size_t matrix_size;
    std::cin >> matrix_size;
    std::vector<std::vector<bool>> matrix(matrix_size,
                                          std::vector<bool>(matrix_size, 0));
    for (size_t i = 0; i < matrix.size(); ++i)
    {
        for (size_t j = 0; j < matrix[i].size(); ++j)
        {
            size_t value;
            std::cin >> value;
            matrix[i][j] = value != 0 ? true : false;
        }
    }

    fix_matrix(matrix);
```

```
    print_matrix(matrix);

    return 0;
}
```

▷

Problem 2.2.7 *Consider a real vector space of polynomials of two variables with degree at most* 2013 *and its subspace* V *consisting of polynomials with the property that*

$$\oint_{x^2+y^2=R^2} f(x,y)ds = 0$$

for all real numbers R*. Find* $\dim(V)$*.*

◁ Denote $N = 2013$ and $C_R = \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 = R^2\}$. If we denote the homogeneous part of f of degree k by f_k, we see that

$$I(R) := \int_{C_R} f(x,y)\,ds = \int_{C_R} \sum_{k=0}^{N} f_k(x,y)ds = \sum_{k=0}^{N} \int_{C_R} f_k(x,y)ds = \sum_{k=0}^{N} \int_{C_1} f_k(x,y)ds \cdot R^{k+1},$$

so the integral of each homogeneous part must vanish.

By symmetry, the integral of each monomial $x^\alpha y^\beta$ vanishes when α or β is odd, and is strictly positive when α and β are both even.

So for odd k, the integral vanishes for all monomials of degree k, and for even $k = 2m$, we have m monomials where the exponent of x and y is odd, so their integral vanishes, and for $\mu = 0, \ldots, m-1$, we have a homogeneous polynomial $x^{2\mu}y^{2(m-\mu)} - c_\mu \cdot x^{2m}$ of degree k whose integral vanishes, and these k homogeneous polynomials are linearly independent.

So overall, we lose one dimension for every even $0 \le k \le N$, whence

$$\dim V = \sum_{k=0}^{N}(k+1) - \left(\left\lfloor \frac{N}{2} \right\rfloor + 1\right) = \frac{(N+1)(N+2)}{2} - \left\lfloor \frac{N+2}{2} \right\rfloor.$$

▷

2.3 Exam on 02.06.2013

Problem 2.3.1 *Compute* $\prod_{k=1}^{\infty} \cos(x \cdot 2^{-k})$

◁ For $x = 0$ the answer is trivially 1. If $x \neq 0$, then note that

$$\prod_{k=1}^{n} \cos(x \cdot 2^{-k}) = \frac{1}{2^n \sin(x \cdot 2^{-n})} \prod_{k=1}^{n} 2^n \cos(x \cdot 2^{-k}) \sin(x \cdot 2^{-n}) = \frac{\sin x}{2^n \sin(x \cdot 2^{-n})}$$

so

$$\prod_{k=1}^{\infty} \cos(x \cdot 2^{-k}) = \lim_{n\to\infty} \prod_{k=1}^{n} \cos(x \cdot 2^{-k}) = \lim_{n\to\infty} \frac{\sin x}{2^n \sin(x \cdot 2^{-n})} = \lim_{n\to\infty} \frac{x}{2^n x \cdot 2^{-n}} = 1$$

▷

Problem 2.3.2 *Compute the rank of* $n \times n$ *matrix* A *such that* $A_{i,j} = (i-j)^2$.

◁ We claim that the determinant of the matrix M is given by $M_{i,j} = (a_i + b_j)^m$ where $i, j \in \mathbb{N}_n$ is 0 if $n > m + 1$. We have

$$\begin{aligned}
\det(M) &= \begin{vmatrix} (a_1+b_1)^m & (a_1+b_2)^m & \dots & (a_1+b_n)^m \\ (a_2+b_1)^m & (a_2+b_2)^m & \dots & (a_2+b_n)^m \\ \dots & \dots & \dots & \dots \\ (a_n+b_1)^m & (a_n+b_2)^m & \dots & (a_n+b_n)^m \end{vmatrix} \\
&= \begin{vmatrix} \sum_{k_1=0}^{m} \binom{m}{k_1} a_1^{k_1} b_1^{m-k_1} & \sum_{k_1=0}^{m} \binom{m}{k_1} a_1^{k_1} b_2^{m-k_1} & \dots & \sum_{k_1=0}^{m} \binom{m}{k_1} a_1^{k_1} b_n^{m-k_1} \\ \sum_{k_2=0}^{m} \binom{m}{k_2} a_2^{k_2} b_1^{m-k_2} & \sum_{k_2=0}^{m} \binom{m}{k_2} a_2^{k_2} b_2^{m-k_2} & \dots & \sum_{k_2=0}^{m} \binom{m}{k_2} a_2^{k_2} b_n^{m-k_2} \\ \dots & \dots & \dots & \dots \\ \sum_{k_n=0}^{m} \binom{m}{k_n} a_n^{k_n} b_1^{m-k_n} & \sum_{k_n=0}^{m} \binom{m}{k_n} a_n^{k_n} b_2^{m-k_n} & \dots & \sum_{k_n=0}^{m} \binom{m}{k_n} a_n^{k_n} b_n^{m-k_n} \end{vmatrix} \\
&= \sum_{k_1=0}^{m} \sum_{k_2=0}^{m} \dots \sum_{k_n=0}^{m} \binom{m}{k_1} \binom{m}{k_2} \dots \binom{m}{k_n} \begin{vmatrix} a_1^{k_1} b_1^{m-k_1} & a_1^{k_1} b_2^{m-k_1} & \dots & a_1^{k_1} b_n^{m-k_1} \\ a_2^{k_2} b_1^{m-k_2} & a_2^{k_2} b_2^{m-k_2} & \dots & a_2^{k_2} b_n^{m-k_2} \\ \dots & \dots & \dots & \dots \\ a_n^{k_n} b_1^{m-k_n} & a_n^{k_n} b_2^{m-k_n} & \dots & a_n^{k_n} b_n^{m-k_n} \end{vmatrix} \\
&= \sum_{k_1=0}^{m} \sum_{k_2=0}^{m} \dots \sum_{k_n=0}^{m} \binom{m}{k_1} \binom{m}{k_2} \dots \binom{m}{k_n} a_1^{k_1} a_2^{k_2} \dots a_n^{k_n} b_1^{m} b_2^{m} \dots b_n^{m} \begin{vmatrix} b_1^{-k_1} & b_2^{-k_1} & \dots & b_n^{-k_1} \\ b_1^{-k_2} & b_2^{-k_2} & \dots & b_n^{-k_2} \\ \dots & \dots & \dots & \dots \\ b_1^{-k_n} & b_2^{-k_n} & \dots & b_n^{-k_n} \end{vmatrix}
\end{aligned}$$

Denote

$$M_{k_1,k_2,\dots,k_n} = \begin{vmatrix} b_1^{-k_1} & b_2^{-k_1} & \dots & b_n^{-k_1} \\ b_1^{-k_2} & b_2^{-k_2} & \dots & b_n^{-k_2} \\ \dots & \dots & \dots & \dots \\ b_1^{-k_n} & b_2^{-k_n} & \dots & b_n^{-k_n} \end{vmatrix}$$

Note that if $k_i = k_j$ for some $i, j \in \mathbb{N}_n$, then the determinant $M_{k_1,k_2,\dots,k_n}$ have two equal columns and therefore equals zero. Now note that if $n > m + 1$ there always exist $i, j \in \mathbb{N}_n$ such that $k_i = k_j$, so $M_{k_1,k_2,\dots,k_n} = 0$ for all tuples $(k_1, k_2, \dots, k_n) \in \{0, \dots, m\}^n$ and as the consequence $\det(M) = 0$.

Now we return to the original problem. For $n = 1, 2, 3$ we have the following matrices

$$A = (0) \qquad A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 1 & 4 \\ 1 & 0 & 1 \\ 4 & 1 & 0 \end{pmatrix}$$

respectively. Their ranks are 0, 2 and 3 respectively. If $n > 3$ by previous claim all minors of A of size greater than 3 are zero, hence $\text{rank}(A) \leq 3$. Since $n > 3$, then A contains a minor

$$\begin{pmatrix} 0 & 1 & 4 \\ 1 & 0 & 1 \\ 4 & 1 & 0 \end{pmatrix}$$

whose rank is 3, so $\text{rank}(A) = 3$. Finally

$$\text{rank}(A) = \begin{cases} 0 & \text{if} \quad n = 1 \\ 2 & \text{if} \quad n = 2 \\ 3 & \text{if} \quad n > 2 \end{cases}$$

▷

Problem 2.3.3 *Given set $A = \{1, 2, \ldots, 256\}$ find the size of maximal subset $A' \subset A$ such that it doesn't contain elements x and y with the property $x = 2y$.*

◁ Let's split A into subsets $A_0, \ldots, A_8$ where A_i contains all numbers in A with exactly i powers of 2 in its factorization. Then $A_{i+1} \subset 2A_i$ and since the sets diminish in size the maximal subset would be $A_0 \cup A_2 \cup \ldots \cup A_8$. It remains to compute the cardinality of A_i for each even i. Since 2^8 is the biggest number in A, then $|A_8| = 1$. For the remaining values of i we have $|A_i| = \frac{256}{2^{i+1}}$, so the desired cardinality is

$$N = 1 + \sum_{0 \leq 2i \leq 7} \frac{256}{2^{2i+1}} = 171$$

▷

Problem 2.3.4 *There are n uniformly disturbed points on a circle. Find the probability that they belong to some semicircle.*

◁ Let $A_n(x)$ denote the event that all points fit into the arc of length $2\pi x$ and belong to some semicircle. Let $B_n(x)$ denote the event that the minimal arc containing n points is of length $2\pi x$, then

$$\mathbb{P}(A_{n+1}(x)) = \int_0^x \mathbb{P}(A_{n+1}(x)|B_n(t))d(\mathbb{P}(A_n(t)))$$

Now we need to find $\mathbb{P}(A_{n+1}(x)|B_n(t))$. If the minimal arc containing n points is of the length $2\pi x$, then in order to have $n + 1$ points inside the arc of length at most $2\pi x$ we need that the $n + 1$-th point to be at most $2\pi(x - t)$ radians to the left from the most left point or to the right from the most right point of the first n points. Therefore $\mathbb{P}(A_{n+1}(x)|B_n(t)) = 2x - t$ and we get

$$\mathbb{P}(A_{n+1}(x)) = \int_0^x (2x - t)d(\mathbb{P}(A_n(t))) = x\mathbb{P}(A_n(x)) + \int_0^x \mathbb{P}(A_n(t))dt$$

Note that $\mathbb{P}(A_1(x)) = 1$. We claim that $\mathbb{P}(A_n(x)) = nx^{n-1}$. Base of induction is obvious. Step of induction: assume $\mathbb{P}(A_n(x)) = nx^{n-1}$, then

$$\mathbb{P}(A_{n+1}(x)) = x\mathbb{P}(A_n(x)) + \int_0^x \mathbb{P}(A_n(t))dt = xnx^{n-1} + \int_0^x nt^{n-1}dt = (n+1)x^n$$

Thus we proved that $\mathbb{P}(A_n(x)) = nx^{n-1}$. The desired probability is

$$\mathbb{P}(A_n(0.5)) = \frac{n}{2^{n-1}}$$

▷

Problem 2.3.5 *We say that a two-dimensional array $A[1\ldots n][1\ldots n]$ of real numbers is increasing, if $A[k][l] \geq A[i][j]$ provided $i \leq k$ and $j \leq l$. Prove that there is no algorithm that finds a real number X in the increasing array $A[1\ldots n][1\ldots n]$ in less than n comparisons.*

◁ Take an n tuple of distinct integers $(a_1, \ldots, a_n)$ and place them on the secondary diagonal of array A. If $i+j < n$ we set $A[i][j] = \min(a_1, \ldots, a_n)$ and if $i+j > n$ we set $A[i][j] = \max(a_1, \ldots, a_n)$. It is clear that A is an increasing array in the sense given above. Since the knowledge of the value of the element on the secondary diagonal doesn't say anything about values of other elements on the secondary diagonal, and these values could be arbitrary we need at least iterate through the whole secondary diagonal. Since it is of length n, then we need at least n comparisons. ▷

Problem 2.3.6 *A linear map of n-dimensional space has $n+1$ eigenvector, such that any n of them is linearly independent. Describe all such matrices.*

◁ By assumption the linear map A has a $n+1$ eigenvectors, say $(e_i)_{i\in\mathbb{N}_{n+1}}$ with eigenvalues $(\lambda_i)_{i\in\mathbb{N}_{n+1}}$. By assumption the vectors $(e_i)_{i\in\mathbb{N}_n}$ are linearly independent. Since A is defined on n dimensional linear space, then $(e_i)_{i\in\mathbb{N}_n}$ is a basis. Hence there exist coefficients $(\alpha_i)_{i\in\mathbb{N}_n}$ such that $e_{n+1} = \sum_{i=1}^n \alpha_i e_i$. By assumption $(e_1, \ldots, e_{m-1}, e_{n+1}, e_{m+1}, \ldots, e_n)$ is a system of linearly independent vectors for all $m \in \mathbb{N}_n$. Since A is defined on n dimensional linear space, then this system of vectors is a basis. Therefore the determinant

$$\begin{vmatrix} 1 & 0 & \ldots & 0 & \alpha_1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & \alpha_2 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & \alpha_{m-1} & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & \alpha_m & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & \alpha_{m+1} & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & \alpha_n & 0 & \ldots & 1 \end{vmatrix}$$

is non-zero. Clearly, this determinant equals to α_m, so we get $\alpha_m \neq 0$ for all $m \in \mathbb{N}_n$. Since e_{n+1} is an eigenvector, then $\lambda_{n+1} = \lambda_k$ for some $k \in \mathbb{N}_n$. Thus $A(e_{n+1}) = \lambda_k e_{n+1}$. Recall the formula for e_{n+1}, and the fact that $A(e_n) = \lambda_n e_n$. Therefore we get $\sum_{i=1}^n \alpha_i(\lambda_i - \lambda_k)e_i = 0$. Since $(e_i)_{i\in\mathbb{N}_n}$ is a basis $\alpha_i(\lambda_i - \lambda_k) = 0$ for all $i \in \mathbb{N}_n$. As we showed earlier $\alpha_i \neq 0$ for all $i \in \mathbb{N}_n$, so for $\lambda_i = \lambda_k$ for all $i \in \mathbb{N}_n$. In other words all eigenvalues of A are equal. It remains to show that any matrix with n equal eigenvalues has $n+1$ eigenvector with the property that any n of them are linearly independent. Indeed we can take first n vectors $(e_i)_{i\in\mathbb{N}}$ to be eigenvectors corresponding to each eigenvalue and set $e_{n+1} = \sum_{i=1}^n e_i$. By argument with determinant given above it follows that any n vectors of the system $(e_i)_{i\in\mathbb{N}}$ are linearly independent. ▷

Problem 2.3.7 *Compute*

$$\sum_{n=1}^{\infty} \frac{f(n)}{n(n+1)}$$

where $f(n)$ is the number of 1's in the binary representation of n.

◁ For each $i \in \mathbb{N}^*$ by $f_i(n)$ we denote i-th digit in the binary exapnsion of n, then for any $n \in \mathbb{N}$ we have $n = \sum_{i=0}^{\infty} f_i(n)2^i$. Note that $f_i(n) = 1$ iff $n = k + 2^i + j2^{i+1}$ for some $k \in \{0, \dots, 2^i - 1\}$ and $j \in \mathbb{N}$. Now we compute the desired sum

$$\sum_{n=1}^{\infty} \frac{f(n)}{n(n+1)} = \sum_{n=1}^{\infty} \frac{1}{n(n+1)} \sum_{i=0}^{\infty} f_i(n)$$

$$= \sum_{i=0}^{\infty} \sum_{n=1}^{\infty} \frac{f_i(n)}{n(n+1)} = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{2^i-1} \frac{1}{(k + 2^i + j2^{i+1})(k + 1 + 2^i + j2^{i+1})}$$

$$= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{2^i-1} \left(\frac{1}{k + 2^i + j2^{i+1}} - \frac{1}{k + 1 + 2^i + j2^{i+1}} \right) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \left(\frac{1}{2^i + j2^{i+1}} - \frac{1}{2^i + 2^i + j2^{i+1}} \right)$$

$$= \sum_{i=0}^{\infty} \frac{1}{2^i} \sum_{j=0}^{\infty} \left(\frac{1}{1 + 2j} - \frac{1}{2 + 2j} \right) = \left(\sum_{i=0}^{\infty} \frac{1}{2^i} \right) \sum_{j=0}^{\infty} \frac{(-1)^j}{j+1} = \frac{1}{1 - 1/2} \sum_{j=0}^{\infty} (-1)^j \int_0^1 x^j dx =$$

$$= 2 \int_0^1 \sum_{j=0}^{\infty} (-1)^j x^j dx = 2 \int_0^1 \sum_{j=0}^{\infty} (-x)^j dx = 2 \int_0^1 \frac{1}{1+x} dx = 2 \ln 2$$

▷

2.4 Exam on 01.06.2012

Problem 2.4.1 *A recurrence sequence defined as follows $x_1 = a$, $x_2 = b$ and $x_{n+1} = \frac{1}{2}(x_n + x_{n-1})$ for $n > 1$. Find $\lim\limits_{n\to\infty} x_n$.*

◁ Denote $y_n = x_{n+1} - x_n$, then $y_1 = b - a$ and $y_n = -\frac{1}{2}y_{n-1}$. Therefore $y_n = \left(-\frac{1}{2}\right)^{n-1}(b-a)$. Note that

$$x_n - x_1 = \sum_{k=1}^{n-1} y_k = \sum_{k=1}^{n-1} \left(-\frac{1}{2}\right)^{k-1} (b-a)$$

so

$$\lim_{n\to\infty} x_n = \lim_{n\to\infty} \left(x_1 + \sum_{k=1}^{n-1} \left(-\frac{1}{2}\right)^{k-1} (b-a) \right) = a + (b-a) \sum_{k=1}^{\infty} \left(-\frac{1}{2}\right)^{k-1} = a + \frac{2}{3}(b-a) = \frac{a+2b}{3}$$

▷

Problem 2.4.2 *Compute $\int_0^1 \varphi(x)\varphi'(x)dx$ where $\varphi(x) = \sum_{k=1}^{\infty} 2^{-2\lfloor \log_2 k \rfloor} x^k$.*

◁ Direct computation shows

$$I := \int_0^1 \varphi(x)\varphi'(x)dx = \frac{1}{2}\int_0^1 d(\varphi^2(x)) = \frac{1}{2}(\varphi^2(1) - \varphi^2(0))$$

Clearly, $\varphi(0) = 0$, so $I = \frac{1}{2}\varphi^2(1)$. Note that

$$\varphi(1) = \sum_{k=1}^{\infty} 2^{-2\lfloor \log_2 k \rfloor} = \sum_{n=0}^{\infty} \sum_{i=2^n}^{2^{n+1}} 2^{-2n} = \sum_{n=0}^{\infty} 2^{-2n} \cdot 2^n = \sum_{n=0}^{\infty} 2^{-n} = 2$$

so $I = \frac{1}{2}\varphi^2(1) = 2$ ▷

Problem 2.4.3 *Consider all nonempty subsets of $\mathbb{N}_n$. For each subset multiply its elements. Find the sum of inverses of these $2^n - 1$ products.*

◁ This sum can be written as

$$\prod_{k=1}^{n} \left(1 + \frac{1}{k}\right) - 1 = \frac{2}{1} \cdot \frac{3}{2} \cdot \ldots \cdot \frac{n+1}{n} - 1 = n + 1 - 1 = n$$

▷

Problem 2.4.4 *Wolf Palme and Ravi Shankar toss a fair coin. Wolf tosses the coin n times, and Ravi $n+1$ times. Find the probability that Ravi had more heads than Wolf.*

◁ By ξ_i and η_i we denote the number of heads got by Ravi and Wolf at i-th toss respectively. Then we set $X_k = \sum_{i=1}^{k} \xi_i$ and $Y_k = \sum_{i=1}^{k} \eta_i$. These random variables are numbers of heads got by Ravi and Wolf after k tosses respectively. Denote $p = \mathbb{P}(\{X_n > Y_n\})$, then by symmetry $\mathbb{P}(\{X_n < Y_n\}) = p$ and, as the consequence, $\mathbb{P}(\{X_n = Y_n\}) = 1 - 2p$. Clearly, $X_{n+1} > Y_n$ in two cases $X_n > Y_n$ or $X_n = Y_n$ and on $\xi_{n+1} = 1$. Therefore

$$\mathbb{P}(\{X_{n+1} > Y_n\}) = \mathbb{P}(\{X_n > Y_n\}) + \mathbb{P}(\{\xi_{n+1} = 1 | X_n = Y_n\})$$

$$= \mathbb{P}(\{X_n > Y_n\}) + \mathbb{P}(\{\xi_{n+1} = 1\}) \cdot \mathbb{P}(\{X_n = Y_n\}) = p + \frac{1}{2}(1 - 2p) = \frac{1}{2}$$

▷

Problem 2.4.5 *Prove that each uncountable set of positive numbers admits a countable subset with infinite sum.*

◁ Let A be an uncountable set of positive numbers. Denote $B_n = \{x : x > n^{-1}\}$, $A_n = A \cap B_n$, then $A = \cup_{n=1}^{\infty} A_n$. Since A is uncountable and represented as countable union of sets, then there exists at least one $m \in \mathbb{N}$ such that A_m is uncountable. Take any sequence $(a_i)_{i\in\mathbb{N}} \subset A_m \subset A$. Since for all $i \in \mathbb{N}$ we have $a_i \geq m^{-1}$, then the series $\sum_{i=1}^{\infty} a_i$ diverges. ▷

Problem 2.4.6 *Give an algorithm that checks if an array of n integers is a permutation of integers $1, \ldots, n$. Time complexity $O(n)$.*

◁ We treat the array as permutation. Thus we need to iterate through all its elements and each time we meet a new element we consider it as the begining of new cycle. As we iterate through the cycle we set respective values in array to -1 to indicate that this element was visited. We run through this hypothetical cycle until we meet its begining or an already iterated element. In the latter case the array is not a permutation.

```
#include <iostream>
#include <vector>

bool is_permutation_(std::vector<int> integers)
{
    // dummy check for not being a permutation
    for (int index = 0; index < integers.size(); ++index)
    {
        if (integers[index] > integers.size() || integers[index] < 1)
        {
            return false;
        }
        else
        {
            integers[index]--;
        }
    }

    for (int index = 0; index < integers.size(); ++index)
    {
        // we met not iterated element
        if (integers[index] >= 0)
        {

            int current_position = index;
            int next_position = integers[current_position];
            // run through the cycle and fill it with -1
            do
            {
                integers[current_position] = -1;
                if (next_position == index)
```

```
                {
                    //we found begining of our cycle
                    break;
                }
                if (next_position < 0)
                {
                    //we met another cycle!
                    return false;
                }
                current_position = next_position;
                next_position = integers[current_position];
            } while (true);
        }
    }
    return true;
}

int main()
{
    size_t array_size;
    std::cin >> array_size;
    std::vector<int> integers(array_size, 0);
    for (int i = 0; i < integers.size(); ++i)
    {
        std::cin >> integers[i];
    }

    std::cout << (
        is_permutation_(integers) ?
        "is permutation" :
        "not a permutation"
    );

    std::cin >> array_size;
    return 0;
}
```

▷

Problem 2.4.7 *Let $(A_i)_{i\in\mathbb{N}_n}$ be a family of finite sets. Prove that the matrix $A = (|A_i \cap A_j|)_{i,j\in\mathbb{N}_n}$ is positive semidefinite.*

◁ Denote $A = \cup_{i=1}^{n} A_i$. For each $x \in A$ define matrix B_x such that

$$(B_x)_{i,j} = \begin{cases} 1 & \text{if} \quad x \in A_i \cap A_j \\ 0 & \text{if} \quad x \notin A_i \cap A_j \end{cases}$$

We claim that each B_x is positive definite. Fix $x \in A$. Note that $(B_x)_{i,i} = 0$, then $x \notin A_i$ and as the consequence, for all $j \in \mathbb{N}_n$ we have $(B_x)_{i,j} = (B_x)_{j,i} = 0$. We call this property (S). Denote

I_x the set of indices for which $(B_x)_{i,i} \neq 0$. Now rearragnge enumeration of sets $(A_i)_{i\in\mathbb{N}_n}$ such that elements of I_x goes first. After rearrangement we get the matrix B'_x. Since B_x is symmetric and has property (S) then matrix B'_x will look like

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

Obviously all its minors are zero, so B'_x is positive semidefinite. Since after rearrangement positive semidefiniteness is preserved, then B_x is also positive semidefinite. Note that $A_x = \sum_{x\in A} B_x$, so A is positive semidefinite as sum of positive semidefinite matrices. $\triangleright$

2.5 Exam on 09.06.2013

Problem 2.5.1 *A sequence* $(a_n)_{n\in\mathbb{N}}$ *is defined recursively*

$$a_0 = 1 \quad a_{n+1} = \frac{a_n}{1 + na_n}$$

Find a closed form for a_n*.*

◁ Denote $b_n = a_n^{-1}$, then $b_0 = 1$ and $b_{n+1} = b_n + n$. So

$$b_n - b_0 = \sum_{k=1}^{n-1}(b_{k+1} - b_k) = \sum_{k=1}^{n-1} k = \frac{1}{2}n(n-1)$$

and finally we get

$$a_n = b_n^{-1} = \left(b_0 + \frac{1}{2}n(n-1)\right)^{-1} = \frac{2}{n^2 - n + 2}$$

▷

Problem 2.5.2 *Assume we are given a set* $A = \{1, \ldots, n\}$*. Find the probability that* $\bigcap_{i=1}^{k} A_i = \varnothing$ *for* k *randomly chosen subsets* $A_1, \ldots, A_n$ *of* A*.*

◁ Each subset of A can be represented as binary vector of length n, hence a group of k sets can be represented as $k \times n$ binary matrix. There are 2^{nk} such matrices. We are interested in groups of sets with empty intersection. These are matrices without columns of 1's. There are $2^k - 1$ columns which doesn't consist of 1's. Since we have n columns, the total number of matrices in question is $(2^k - 1)^n$. The desired probability is

$$\frac{(2^k - 1)^n}{2^{kn}} = (1 - 2^{-k})^n$$

▷

Problem 2.5.3 *Give an algorithm that finds the longest subarray of binary array of length* n *with equal number of units and zeros. Restrictions:* $O(n)$ *time complexity,* $O(n)$ *memory.*

◁ Let's change all zeros in the original array to -1. Call this array b. Now we need to find a subarray of b with zero sum of elements. Now consider the graph of the function $F(i) = n + H(i)$ where $H(i)$ is the sum of the first i elements of b. For each $h \in [0, 2n+1]$ we can draw a horizontal line on the height h. If it intersects the graph of F, then consider abscissas of the most left and the most right points of intersections. Call them f_h and l_h, then $b[f_h \ldots l_h]$ is a subarray with zero sum. If there is no intersections with horizontal lines we set $f_h = l_h = -1$. It remains to iterate through all $h \in [0, 2n+1]$ and find h with maximal difference $l_h - f_h$ provided l_h and f_h are positive.

```
#include <algorithm>
#include <iostream>
#include <vector>

std::pair<size_t, size_t> max_balanced_subarray_edges(
```

```
    const std::vector<bool>& values)
{
    std::vector<int> height(values.size() + 1, values.size());
    for (size_t index = 1; index < height.size(); ++index)
    {
        height[index] = height[index - 1] +
            (values[index - 1] ? 1 : -1);
    }

    int flag = -1;

    std::vector<int> first_occurence_of_height(
        2 * values.size() + 1, flag);
    std::vector<int>  last_occurence_of_height(
        2 * values.size() + 1, flag);

    for (size_t index = 0; index < height.size(); ++index)
    {
        if (first_occurence_of_height[height[index]] == flag)
        {
            first_occurence_of_height[height[index]] = index;
        }
        else
        {
            last_occurence_of_height[height[index]] = index;
        }
    }

    size_t begin, end;
    size_t max_length = 0;

    for (size_t height = 0; height < 2 * values.size() + 1; ++height)
    {
        if (first_occurence_of_height[height] != flag &&
            last_occurence_of_height[height] != flag)
        {
            if (last_occurence_of_height[height] -
                first_occurence_of_height[height] > max_length)
            {
                max_length = last_occurence_of_height[height] -
                             first_occurence_of_height[height];
                begin = first_occurence_of_height[height];
                end = last_occurence_of_height[height];
            }
        }
    }
```

```
    return std::make_pair(begin, end - 1);
}

int main()
{
    size_t array_size;
    std::cin >> array_size;
    std::vector<bool> values(array_size, false);
    for (size_t index = 0; index < values.size(); ++index)
    {
        int value;
        std::cin >> value;
        values[index] = value != 0 ? true : false;
    }

    std::pair<size_t, size_t> edges = max_balanced_subarray_edges(values);
    std::cout << edges.first << ' ' << edges.second;

    return 0;
}
```

▷

Problem 2.5.4 *Find all* $m \in \{1, \ldots, 10\}$ *such that*

$$I_m = \int_0^{2\pi} \cos(x) \cdot \ldots \cdot \cos(mx) dx \neq 0$$

◁ Let $C(l, m)$ denote the number or representations of l in the form $\sum_{r=1}^{m} \alpha_r$ where $\alpha_r \in \{-r, r\}$. Then

$$I_m = \int_0^{2\pi} \prod_{k=1}^{m} \cos(kx) dx = \int_0^{2\pi} \prod_{k=1}^{m} \frac{1}{2}(e^{ikx} + e^{-ikx}) dx = 2^{-m} \int_0^{2\pi} \sum_{l=-m}^{m} C(l, m) e^{ilx} dx$$

$$= 2^{-m} \sum_{l=-m}^{m} C(l, m) \int_0^{2\pi} e^{ilx} dx = 2^{-m} \sum_{l=-m}^{m} C(l, m) 2\pi \delta_{l,0} = 2^{1-m} \pi C(0, m)$$

Therefore $I_m \neq 0$ iff $C(0, m) \neq 0$, so we had to decide for each $m \in \mathbb{N}_{10}$ whether 0 is representable in the form $\sum_{r=1}^{m} \alpha_r$ for some $\alpha_r \in \{-r, r\}$. Note that $\sum_{r=1}^{m} \alpha_r$ have the same parity as $S_m := \sum_{r=1}^{m} r$. Since $S_1, S_2, S_5, S_6, S_9, S_{10}$ are odd then for $m \in \{1, 2, 5, 6, 9, 10\}$ we have $C(0, m) = 0$ and as the consequence $I_m = 0$. For the remaining values of m we can present at least one representation: $1 + 2 - 3 = 0$ for $m = 3$, $1 - 2 - 3 + 4 = 0$ for $m = 4$, $1 + 2 - 3 - 4 + 5 + 6 - 7 = 0$ for $m = 7$, $1 + 2 + 3 - 4 + 5 - 6 + 7 - 8 = 0$ for $m = 8$. Hence, for $m \in \{3, 4, 7, 8\}$ we have $C(0, m) \neq 0$ and $I_m \neq 0$ too. ▷

Problem 2.5.5 *Let* G *be a nonoriented graph without loops. Prove its adjacency matrix has negative a eigenvalue.*

◁ Since graph is non oriented then its adjacency matrix A is a real symmetric matrix. Hence all eigenvalues of A are real. Since G has no loops, then its main diagonal contains only zeros, hence trace of A is zero. Since trace also equals to the sum of all eigenvalues, then we have two separate cases. First case A has positive and negaitve eigenvalues, Second case all eigenvalues of A are zero. In the first case we are done, in the latter case A is zero, which means that G has no edges. ▷

Problem 2.5.6 *Consider infinite two dimensional array $(a_{i,j})_{i,j\in\mathbb{N}}$ with the property that each its element repeats exactly 8 times. Prove that there exist m and n such that $a_{m,n} > mn$.*

◁ Assume $a_{m,n} \leq mn$ for all $m, n \in \mathbb{N}$. For any fixed $k \in \mathbb{N}$ consider set $A_k = \{(m, n) : mn \leq k\}$. Clearly $|A_k| \leq \int_2^k \frac{k}{x} dx = k \ln(k/2)$. From assumption $|A_k| \geq 8k$ for all $k \in \mathbb{N}$. Therefore $8k \leq k \ln(k/2)$, for all $k \in \mathbb{N}$. Contradiction, hence there exist $(m, n) \in \mathbb{N}^2$ such that $a_{m,n} > mn$. ▷

Problem 2.5.7 *Assume we are given a binary matrix such that each its row is zero or contains exactly one group of 1's followed one after another. Prove that determinant of such matrix is 0 or 1 or −1.*

◁ By basic properties of determinants we can arrange rows of the matrix A to get upper trapezoid form B. During rearrangement, if we have a group of rows where 1's start at the same index we place on the top a row with smaller length of group of 1's. After such rearrangemnt of rows determinant of original matrix can only change its sign. Thus it is enough to prove the result for matrices of the upper trapezoid form with the property that rows having the same index of the first 1 sorted from top to bottm by increase of the length of the group of 1's. In other words we need to prove that determinant of the matrix of the form like

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

is 0 or ± 1.

We prove this claim by induction on size of the matrix. Basis of induction: if $n = 1$ then matrix is either (0) or (1), so its determinant is 0 or 1. Induction step: assume claim is valid for $n = k$. Consider arbitrary $(k + 1) \times (k + 1)$-matrix B of the form described above. If its first column is zero, then $\det(B) = 0$. Otherwise, first element of the first row is 1. We use it as pivot (if necessary) to make zeros in the first column under it. This is possible since subtraction of rows doesn't change the deteminant. Thanks to arrangement of rows after such subtractions we get the

matrix B' of the same form. For example, for matrix described above we get

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

Now we expand this deteminant in the first columnn and see that $\det(B') = 1M_{1,1} = M_{1,1}$. After sutable rearrangement of rows the minor $M_{1,1}$ of matrix B' becomes a determinant of the $k \times k$ matrix of the form described above. By inductive assummption we have $M_{1,1}$ is either 0 or ± 1. Thus, $\det(B) = \det(B') = \det(M_{1,1})$ is either 0 or ± 1. Induction step finished. $\triangleright$

2.6 Exam on 19.02.2014

Problem 2.6.1 *Find all 3×3 matrices X such that $X^2 + E = 0$.*

◁ Since we consider 3×3 matrices, then $\det(-E) = -1$. Hence, $\det(X)^2 = \det(X)\det(X) = \det(X^2) = \det(-E) = -1 < 0$. Contradiction, hence such matrix doesn't exist. ▷

Problem 2.6.2 *During camping among any four guys at least one is acquainted with other three guys. Prove that each participant save at most three is acquainted with others.*

◁ We call participants that are not acquainted with all others non-friendly with respect to his group. The remaining participants are called friendly. We perform proof by induction on the number of participants n. Base of induction for $n = 4$ is obvious. Assume the claim is true for any group of at most n participants. Consider group G of $n + 1$ participants. Take any subgroup G' of n participants. From statement of the problem it is clear that there cannot exist a unique non-friendly participant with respect to G', because all other participants must be acquainted with him, and therefore non-friendly participant must be friendly. Therefore from here we have two cases.

First case: there are two non-friendly participants in G'. Call them a and b. Clearly all participants from $G' \setminus \{a, b\}$ are acquainted with a and b. Therefore a and b don't know each other. Consider a unique participant $z \in G \setminus G'$ and any participant $p \in G' \setminus \{a, b\}$. Consider group $S = \{a, b, p, z\}$. By assumption we have a participant in S who knows others. Since a and b don't know each other this must be either p or z. In both cases we get that z is acquainted with p. Since p is arbitrary we get that z knows all participants from $G' \setminus \{a, b\}$. If z knows a and b he is friendly in G, so G has two non-friendly participants and inductive step is finished in this subcase. If z does not know a or b, then he is non-friendly in G, but G still has the property that among any 4 participants there is a friendly participant. Hence the claim is proved in the this subcase too.

Second case: there are three non-friendly participants in G'. Call them a, b and c. Clearly all participants from $G' \setminus \{a, b, c\}$ are familiar with a, b and c. Hence among a, b and c there is at least one pair of participants who don't know each other. Consider a unique participant $z \in G \setminus G'$ and consider group $\{z, a, b, c\}$. If there is no acquaintances among a, b and c then from the problem statement it follows that z knows a, b and c. Assume now there are acquaintances among a, b and c. Clearly a, b and c can't know each other because they are non friendly. Clearly there is no participant in $\{a, b, c\}$ that knows the remaining two, because otherwise he will be friendly in G'. Therefore there is only one acquaintance, say a knows b. Again from problem statement there is a participant in $\{z, a, b, c\}$ that knows the remaining participants. The only possibility here is that z knows a, b and c. All in all we proved that z knows a, b and c. Now take any $p \in G' \setminus \{a, b, c\}$ and consider group $S = \{z, p, a, c\}$. Since p knows a and c and c doesn't know a, then from problem statement we have that either z or p knows the remaining participants in S. In both cases z knows p. Since $p \in G' \setminus \{a, b, c\}$ is arbitrary, then z knows all participants in G'. But he also knows a, b and c. So he is friendly in G. Therefore the only non-friendly participants in G are a, b and c. Thus the claim is proved in the second case too and the step of induction is completed. ▷

Problem 2.6.3 *Describe all non-singular real-valued matrices A such that all entries of A and A^{-1} are non negative.*

◁ Assume A is invertible $n \times n$ matrix and A with A^{-1} have non negative entries. Denote $B = A^{-1}$, then $AB = BA = E$. Hence for all $i, k \in \mathbb{N}_n$

$$\sum_{j=1}^{n} A_{i,j} B_{j,k} = \delta_{i,j}$$

Since A and B have non negative entries, then for any $i, j, k \in \mathbb{N}_n$ such that $i \neq k$ we have $A_{i,j} = 0$ or $B_{j,k} = 0$. Assume there exist $i, j_1, j_2 \in \mathbb{N}_n$ such that $j_1 \neq j_2$ and $A_{i,j_1} \neq 0$, $A_{i,j_2} \neq 0$, then for any $k \neq i$ we have $B_{j_1,k} = 0$ and $B_{j_2,k} = 0$. Clearly, $B_{j_1,i} \neq 0$ and $B_{j_2,i} \neq 0$, otherwise B will have a zero row and will be singular. Since $B_{j_1,i} \neq 0$ and $B_{j_2,i} \neq 0$ while $B_{j_1,k} = B_{j_2,k} = 0$ for all $k \neq i$, then B have two linearly dependent rows,hence it is singular. Contradiction, therefore for any $i \in \mathbb{N}_n$ there is unique $\pi(i) \in \mathbb{N}_n$ such that $A_{i,\pi(i)} \neq 0$. Assume $\pi(i_1) = \pi(i_2)$ for some $i_1, i_2 \in \mathbb{N}_n$ such that $i_1 \neq i_2$, then A have two linearly dependent rows, hence A is singular. Contradiction, so $\pi : \mathbb{N}_n \to \mathbb{N}_n$ is an injective function, hence π is a permutation. Thus matrix A is a permutation of columns of a diagonal matrix D. Diagonal elements of D are nonzero, because A is invertible, and positive because A is non-negative.

Conversely, let A be constructed from diagonal matrix $D = \operatorname{diag}(a_1, \ldots, a_n)$ where $a_i > 0$ for all $i \in \mathbb{N}_n$ by permutation of its columns. Denote this permutation by π, then $A(e_{\pi(i)}) = a_i e_i$, where $(e_i)_{i \in \mathbb{N}_n}$ is a standard basis of $\mathbb{R}^n$. It is clear now, that $A^{-1}(e_i) = a_i^{-1} e_{\pi(i)}$. Hence A^{-1} have non negative entries. ▷

Problem 2.6.4 *Give an algorithm that finds the maximal value of sums of subarrays of a given array $a[1 \ldots n]$. Time complexity $O(n)$, memory restriction $O(1)$.*

◁ The maximum is initially zero. Suppose that we've solved the problem for subarray $a[1 \ldots i-1]$ how can we extend that to a solution for the first i elements? The maximum sum in the first i elements is either the maximum sum in the first $i-1$ elements (which we'll call max so far), or it is that of a subvector that ends in position i (which we'll call max ending here). Instead of computing the maximum subvector ending in position I from scratch, we'll use the maximum subvector that ends in position $i - 1$.

```
#include <iostream>
#include <algorithm>
#include <vector>

double maximal_subarray_sum(const std::vector<double>& values)
{
    double max_so_far = 0.;
    double max_ending_here = 0.;
    for (size_t index = 0; index < values.size(); ++index)
    {
        max_ending_here = std::max(max_ending_here + values[index], 0.);
        max_so_far = std::max(max_so_far, max_ending_here);
    }

    return max_so_far;
}
```

```
int main()
{
    size_t size;
    std::cin >> size;
    std::vector<double> values(size);
    for (size_t index = 0; index < values.size(); ++index)
    {
        std::cin >> values[index];
    }

    std::cout << maximal_subarray_sum(values);

    return 0;
}
```

▷

Problem 2.6.5 *There are* 10 *coins. With one weighing one can determine which coin is heavier. Is it possible to find the heaviest coin in* 20 *weighings?*

◁ There are 10! different rearragements of 10 coins. This knowledge is worth of $\lfloor \log_2(10!) \rfloor + 1$ bits, which is 23 bits of information. Each weighing gives us 1 bit of information, hence 20 weighings is not enough. ▷

Problem 2.6.6 *Compute*

$$\int_{\sqrt{\pi/6}}^{\sqrt{\pi/3}} \sin(x^2)dx + \int_{1/2}^{\sqrt{3}/2} \sqrt{\arcsin(x)}dx$$

◁ Note that if f is a strictly increasing continuous function, then

$$\int_a^b f(x)dx + \int_{f(a)}^{f(b)} f^{-1}(x)dx = bf(b) - af(a)$$

Applying this result to $f = \sin(x^2)$ with $a = \sqrt{\pi/6}$ and $b = \sqrt{\pi/3}$ we get

$$\int_{\sqrt{\pi/6}}^{\sqrt{\pi/3}} \sin(x^2)dx + \int_{1/2}^{\sqrt{3}/2} \sqrt{\arcsin(x)}dx = \sqrt{\frac{\pi}{3}} \cdot \frac{\sqrt{3}}{2} - \sqrt{\frac{\pi}{6}} \cdot \frac{1}{2} = \frac{\sqrt{\pi}(\sqrt{6}-1)}{2\sqrt{6}}$$

▷

Problem 2.6.7 *A game consist of equal independent rounds, with probability of win equal to* p*. If a player wins he gets* 1 *dollar, otherwise he pays* 1 *dollar. As soon as he collect* N *dollars he finishes the game. Find the probability that the player will lose all his money if he starts the game with* K *dollars.*

◁ Let $P_k(s)$ be the probaility of player to loose after making at most k moves provided his current current stock is s dollars. Then we have

$$P_k(s) = pP_{k-1}(s+1) + (1-p)P_{k-1}(s-1)$$

with initial conditions $P_0(s) = 0$ for $s > 0$, $P_0(s) = 1$ for $s \leq$ and $P_k(0) = 1$, $P_k(N) = 0$ for all $k \in \mathbb{N}$. Since the number of possible moves is not bounded, we need to consider asymptotical behaviour of the probability, i.e. find the closed form of $P(s) = \lim_{k\to\infty} P_k(s)$. So we have the following recurrence equation

$$P(s) = pP(s+1) + (1-p)P(s-1)$$

with bounddary condition $P(0) = 1$, $P(N) = 0$. This is recurrence equation of the second order with characteristic equation

$$p\lambda - 1 + \frac{1-p}{\lambda} = 0$$

Its roots are $\lambda = 1 - p/p$ and $\lambda = 1$. So

$$P(s) = C_1\left(\frac{1-p}{p}\right)^s + C_2$$

Since $P(0) = 1$, $P(N) = 0$ we get

$$C_1 + C_2 = 1 \qquad C_1\left(\frac{1-p}{p}\right)^N + C_2 = 1$$

so

$$C_1 = \frac{p^N}{p^N - (1-p)^N} \qquad C_2 = -\frac{(1-p)^N}{p^N - (1-p)^N}$$

$$P(s) = \frac{p^N}{p^N - (1-p)^N}\left(\left(\frac{1-p}{p}\right)^s - \left(\frac{1-p}{p}\right)^N\right)$$

The desired probability equals

$$P(K) = \frac{p^N}{p^N - (1-p)^N}\left(\left(\frac{1-p}{p}\right)^K - \left(\frac{1-p}{p}\right)^N\right)$$

▷

Problem 2.6.8 *For a given real number a and an integer $n \geq 0$ by a_n we denote the distantce from a to the closest number of the form $m/2^n$ where m is an integer. Find the maximal possible value of*

$$\sum_{n=0}^{\infty} a_n$$

◁ Clearly, for any real a we have $a_n \leq 2^{-n-1}$ so $\sum_{n=0}^{\infty} a_n \leq \sum_{n=0}^{\infty} 2^{-n-1} = 1$. This is maximal possible value for the sum in question and maximum is attained for $a = 0$. Indeed, $a^* = 0$ and $a_n^* = 2^{-n}$ for $n \in \mathbb{N}$, hence $\sum_{n=0}^{\infty} a_n^* = 0 + \sum_{n=1}^{\infty} 2^{-n} = 1$. ▷