Задача 2

$|x-y| \leq 2^n$. Если $p_1, \cdots, p_k : \forall i = 1, \cdots, k \hookrightarrow (|x-y| \vdots p_i) \wedge (n \leq p_i \leq 2n) \Rightarrow |x-y| \geq p_1 \cdots p_k \geq n^k \Rightarrow 2^n \geq n^k \Rightarrow k \leq \ln 2 \frac{n}{\ln n}$ Учитывая, что при $n \to \infty \hookrightarrow \pi(n) \sim \frac{n}{\ln n}$, получаем оценку на вероятность ошибки

$$P_{err} \leq \frac{k}{\pi(2n) - \pi(n)} \leq \frac{\ln 2}{\dfrac{2 \ln n}{\ln 2n} - 1} = \frac{\ln 2 \cdot \ln 2n}{\ln n - \ln 2} = \ln 2 \frac{\ln 2n}{\ln n/2}$$

$$P_{err} \leq \frac{3}{4} \Rightarrow n \geq 2^{\frac{3+\ln 16}{3-\ln 16}} \approx 2^{25.384} \text{ бит}$$

Такая длины файлов достаточна для справедливости оценки вероятности ошибки. 32 мб равны $2^8 2^6 5^6$ бит $> 2^{26}$ бит $> n_0$, тогда оценка справедлива при такой длине файлов.

Задача 3

(i) Пусть BPP' — класс языков, распознаваемых вероятностной машиной Тьюринга с вероятностью ошибки не больше чем $p < \frac{1}{2}$, работающей полиномиальное в среднем число шагов.
Из определения BPP следует, что $BPP \subset BPP'$, т. к. $p = \frac{1}{3} < \frac{1}{2}$

Основываясь на *inequality for Bernoulli random variables* покажем $BPP' \subset BPP$: пусть МТ $M' \in BPP'$, которая ошибается с вероятностью $P_{er} \leq p : p < \frac{1}{2}$. Построим с ее помощью МТ M, вероятность ошибки которой меньше $\frac{1}{3}$. Для этого запустим машину M' n раз подряд, сохраняя ответы (да или нет на вопрос принадлежности слова языку) и затем вернем в качетсве результата работы машины M произвольный ответ из созраненных n штук. Вероятность ошибки машины M равна числу ошибочных ответов среди n сохраненных ответов, каждый из которых, в свою очередь, является ошибочным с вероятностью p. Тогда вероятность ошибки машины $M = P_{er}(M) \leq \frac{1}{3} \Leftrightarrow$ тому, что число ошибочных ответов меньше либо равно $\frac{n}{3}$ $(*)$. В формуле

$P((\text{number of errors among n results}) \leq (p-e)n) \leq exp(-2e^2 n)$ положим $e = \dfrac{1}{6}$

Т. к. $p < \dfrac{1}{2}$, то $(p-e) < (0.5-e) = \dfrac{1}{3}$. Поскольку в этой формуле самое первое выражение есть вероятность ошибки, как сказано выше $(*)$,

то при $exp(-2e^2n) < \dfrac{1}{3}$ получаем $BPP' \subset BPP$. Для этого достаточно запустить МТ количество раз $n > 18\ln 3$. Например, сто раз. Т. к. на каждом из n шагов МТ M' работала в среднем полиномиально, то это свойство сохранится и для МТ M.
В итоге, $BPP = BPP'$.

(ii) Покажем как осуществить замену полиномиальное в среднем числа шагов работы МТ на строго полиномиальное число шагов с сохранением того, что вероятность ошибки МТ строго меньше $0,5$. Пусть есть МТ, ошибающаяся с вероятностью $e < \frac{1}{2}$, которая работает полиномиальное время с вероятностью p и неполиномиальное с вероятностью $1-p$. Построим по ней МТ' для того же языка, которая работает как МТ, если число шагов меньше либо равно $poly(|w|)$ (w есть слово языка), а иначе прекращает моделирование МТ и возвращает произвольно ответ про принадлежность слова языку ("да"или "нет"). Вероятность ошибки МТ' равна $ep + 0.5(1-p) = 0.5 + p(e - 0.5) < 0.5$. Т. е. новая МТ' разрешает тот же язык за полиномиальное время и принадлежит BPP.

Задача 4

(i) $ABx = Cx \Leftrightarrow (AB - C)x = \bar{0} \Leftrightarrow Mx = \bar{0}$, где $M = AB - C$. Умножив матрицу M на произвольно выбранный вектор x получим n полиномов степени не больше, чем один. Если $rk(M) = n$, то вероятность того, что все полиномы зануляются выбранным вектором по лемме Шварца-Зиппеля равна $\left(\frac{1}{N}\right)^n$, т. к. координаты вектора могут принимать N разных значений и степень многочленов равна 1. Если ранг матрицы системы меньше n, то вероятность обнуления всех многочленов (а это равносильно равенству исходных матриц) будет меньше. Тогда из условия

$$\left(\frac{1}{N}\right)^n < p \Rightarrow N > \frac{1}{\sqrt[n]{p}}$$

(ii) $x^T ABx = x^T Cx$. Перемножив матрицу на произвольно выьранный вектор, после переноса через знак равенства получим полином от n переменных степени не больше второй. Тогда по лемме Шварца-Зиппеля вероятность того, что полином обнуляется выбранным вектором x не более $\dfrac{2}{N}$. Тогда $\frac{2}{N} < p \Rightarrow N > \frac{2}{p}$.

$y^T ABx = y^T Cx$ — Аналогично получаем полином степени не выше второй, но содержащий $2n$ переменных. Тогда вероятность ошибки равна $\frac{2}{2N} = \frac{1}{N}$. $\frac{1}{N} < p \Rightarrow N > \frac{1}{p}$.

Задача 5

(i) Пусть в графе минимальный разрез содержит r ребер. Если у какой-то вершины графа степень меньше r, то, выделив ее и остальные вершины в два дизъюнктных подмножества множества вершин графа, получим разрез с меньшим числом ребер. Противоречие с минимальностью исходного разреза. Тогда степени всех вершин в графе не меньше r. Получаем

$$|E| = \frac{1}{2}\sum_{v\in V} deg(v) \geq \frac{1}{2}|V|k$$

$$P(\text{выбранное ребро в min разрезе}) = \frac{k}{|E|} \leq \frac{k}{0.5|V|k} = \frac{2}{|V|}$$

(ii) Алгоритм выдаст MINCUT, если в процессе его работы никакое ребро из минимального разреза не будет стянуто. Это просходит с вероятностью

$$\left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\left(1-\frac{2}{n-2}\right)\cdots\left(1-\frac{2}{4}\right)\left(1-\frac{2}{3}\right) =$$

$$= \frac{n-2}{n}\cdot\frac{n-3}{n-1}\cdot\frac{n-4}{n-2}\cdots\frac{2}{4}\cdot\frac{1}{3} = \frac{2(n-2)!}{n!} = \frac{2}{n(n-1)}$$

(iii) Вероятность вероятность хотя бы одного ошибочного стягивания при каждом из n^2 независимых выполнений равна $1-\dfrac{2}{n(n-1)} < 1-\dfrac{2}{n^2}$. Тогда вероятность правильно найти минимальный разрез больше

$$1-\left(1-\frac{2}{n^2}\right)^{n^2} > 0.85 \text{ т. к. } 1-\left(1-\frac{2}{n^2}\right)^{n^2} \underset{n\to\infty}{\to} 1-\frac{1}{e^2} \approx 0.865$$

Задача 6

(i) Решим задачу через графы. Для этого преобразуем каждый дизъюнкт $a \vee b \ \equiv\ !a \to b \ \equiv\ !b \to a$ $(*)$.

Построим по 2-КНФ граф : каждая переменная и ее отрицание будут вершинами, а ребра будут соответствовать импликациям (*).

Теперь заметим, что если для какой-то переменной a выполняется, что из нее достижимо $!a$, а из $!a$ достижимо a, то задача решения не имеет, т. к. какое бы значение для переменной a мы бы ни выбрали, всегда получим противоречие ($!a = 1 \wedge a = 1$). Тогда, для того, чтобы 2-КНФ имела решение, необходимо и достаточно, чтобы для любой переменной a вершины a и $!a$ находились в разных компонентах сильной связности построенного графа. Это проверим за $O(|V|+|E|)$ с помощью алгоритма поиска сильно связных компонент и последующего поиска пути между вершинами a и $!a$ (a произвольно). *found at emax*
(ii) Поскольку алгоритм лежит в P и $P \subseteq ZPP \subseteq RP \subseteq BPP$, то он лежит во всех этих вероятностных классах.

Задача 7

(i) База индукции, когда под картой a_{n-1} всего одна карта. Пусть под картой a_{n-1} находятся j карт и каждая их перестановка равновероятна. Тогда вероятность конкретной перестановки под a_{n-1} картой равна $\frac{1}{j!}$. Для вставки новой карты под a_{n-1} карту есть $j+1$ вариант, значит, вероятнсть получить конкретную перестановку равна $\frac{1}{j!} \cdot \frac{1}{j+1} = \frac{1}{(j+1)!}$, т. е. снова все перестановки равновероятны.
(ii) Аналогично пункту (i).
(iii) Пусть T_j — число итераций цикла, когда под картой a_{n-1} ровно j карт. Тогда общее число итераций равно $T = T_1 + T_2 + \cdots + T_{n-2}$ В силу линейности матожидания : $E_T = \sum_{j=1}^{n-2} E_{T_j}$ Если под картой a_{n-1} находятся j карт, получим

$$E_{T_j} = 1 \cdot \frac{j+1}{n} + \cdots + k \cdot \frac{j+1}{n} \left(\frac{n-j-1}{n} \right)^{k-1} = \sum_{k=0}^{\infty} (k+1) \frac{j+1}{n} \left(1 - \frac{j+1}{n} \right)^{k} =$$

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

Тогда

$$E_T = \sum_{j=1}^{n-2} \frac{n}{j+1} = n \ \Theta(\ln n)$$

Т. к. $\ln n - \ln 2 = \int_0^{n-2} \dfrac{dx}{x+2} \leq \sum_{j=1}^{n-2} \dfrac{1}{j+1} \leq \int_0^{n-2} \dfrac{dx}{x+1} = \ln{(n-1)}$