Алгоритмы и модели вычислений. Контрольная 2. 15 мая 2023

◇

Фамилия, имя: Ковалев Алексей

Группа: Б05-105

Преподаватель, ведущий занятия:

Шестаков

- Заполните поля Фамилия, имя, группа и имя преподавателя на обложке.
- Ответы без обоснования не оцениваются.
- При написании контрольной можно пользоваться любыми печатными/рукописными материалами.
- Нельзя пользоваться никакими электронными приспособлениями в том числе для просмотра конспектов или литературы.
- Нельзя пользоваться помощью других студентов, в том числе нельзя передавать печатные или рукописные материалы друг другу во время написания контрольной.
- Пользование электронными приспособлениями, помощью других студентов и списывание приравнивается к нечестному написанию контрольной и наказывается на усмотрение проводящих контрольную вплоть до обнуления работы.
- Всюду, если только явно не указано обратное, предполагается истинность стандартных гипотез $\mathcal{P} \neq \mathcal{NP}, \mathcal{NP} \neq \mathrm{co}\mathcal{NP}$ и тому подобных.
- Можно без доказательства пользоваться результатами, доказанными и разобранными на лекциях и семинарах.
- “Граф” без дополнительных указаний – это всегда неориентированный граф без петель и кратных рёбер, “цикл” – это всегда простой цикл, “путь” – простой путь.
- После написания контрольной вы должны сфотографировать работу на телефон, сформировать .pdf файл, отправить этот файл на почту acm-mipt@yandex.ru, в теме письма указать фамилию и группу. Затем сдать бумажную работу преподавателю.
- Разбалловка контрольной работы будет сообщена после проверки всех работ.

1. Приведите пример двух асимптотически различных неотрицательных функций $f_1 : \mathbb{N} \to \mathbb{R}_+$, $f_2 : \mathbb{N} \to \mathbb{R}_+$, для которых выполняется равенство

$$\mathrm{DTIME}(n^2)/f_1(n) = \mathrm{DTIME}(n^2)/f_2(n)$$

Ответ обоснуйте. (В качестве модели вычисления для DTIME подразумевается обычная многоленточная детерминированная машина Тьюринга).

2. Пусть G – неориентированный граф. Назовём *индексом* пары (s,t) различных вершин графа G количество различных простых путей из s в t в графе G. Рассмотрим язык

$$\text{G-INDEX} = \{(G,s,t,k) : \text{индекс пары } (s,t) \text{ в графе } G \text{ равен } k\}$$

Докажите, что G-INDEX $\in \mathcal{PSPACE}$.

3. Классам языков $\mathbf{X}_1$, $\mathbf{X}_2$ принадлежат все такие языки, что для них существует полиномиальная вероятностная машина Тьюринга с вероятностями, заданными таблицей. Классы $\mathbf{X}_1$, $\mathbf{X}_2$ равны некоторым из известных вам классов. Каким и почему?

$\mathbf{X}_1$	$M(x,r)=1$	$M(x,r)=0$
$x \in L$	$< 1/5$	$> 4/5$
$x \notin L$	$> 4/5$	$< 1/5$

$\mathbf{X}_2$	$M(x,r)=1$	$M(x,r)=0$
$x \in L$	$> 3/4$	$< 1/4$
$x \notin L$	$< 3/4$	$> 1/4$

4. Пусть дана полиномиальная вероятностная машина Тьюринга M, тогда для фиксированного $x \in \{0,1\}^*$ различные случайные строки r определяют различные вычислительные пути M на паре (x,r). Длина вычислительного пути в дереве путей – это длина строки r.
Определим новый сложностной класс $\mathcal{RP}^\dagger$. По определению язык L лежит в $\mathcal{RP}^\dagger$, если существует полиномиальная вероятностная машина Тьюринга M такая, что:

1) если $x \in L$, то по крайней мере половина всех вычислительных путей завершаются в состоянии “принять”;

2) если $x \notin L$, то все вычислительные пути завершаются в состоянии “отвергнуть”.

Здесь не накладывается ограничений на длину вычислительных путей, кроме тех ограничений, что следуют из полиномиальности M.
Докажите, что CLIQUE $\in \mathcal{RP}^\dagger$.

5. Пусть G – ориентированный граф. Назовём пару вершин (s,t) *слабо связанной* в G, если существует вершина v графа G, отличная от s и t, такая что в G существует путь из s в v и путь из t в v.
Рассмотрим язык

$$\text{WEAK-PATH} = \{(G,s,t) : (s,t) \text{ слабо связаны в орграфе } G\}$$

Докажите, что WEAK-PATH $\mathcal{NL}$-полный.

6. Пусть m,n – положительные целые числа, причём n – степень двойки, $m > \log n$. Пусть A – таблица из m строк и n столбцов, причём в каждой клетке (i,j) записано число j (номер столбца). Некоторые клетки таблицы покрашены в красный цвет, некоторые в синий, а некоторые не покрашены. Известно, что в каждом столбце покрашено (суммарно в два цвета) не менее $\log n + 1$ клеток.
Для каждого слова $v \in \{R,B\}^m$ множество S_v есть объединение

- тех элементов первой строки, чей цвет совпадает с первой буквой слова v (R – красный, B – синий),
- тех элементов второй строки, чей цвет совпадает с второй буквой слова v,
- $\ldots$,
- тех элементов строки номер m, чей цвет совпадает с последней буквой слова v.

1) Докажите с помощью вероятностного алгоритма, что существует $v \in \{R,B\}^m$, что $S_v = \{1,\ldots,n\}$.

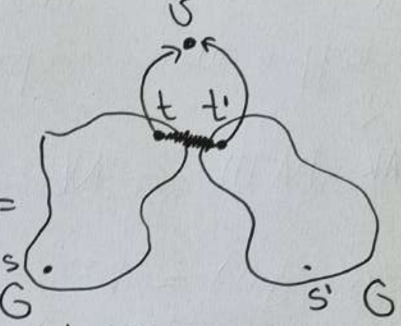
2) Дерандомизируйте алгоритм и получите эффективный детерминированный алгоритм для поиска требуемого v.

№5

DPATH $\in$ NL-complete

DPATH $\leq_{log}$ WEAK-PATH

$f: (G; s; t) \longmapsto (G'; s; s')$, где $G' =$



т.е. G' получен из двух копий G, к которым добавлено ребро $(t; t')$

f вычислима полиномиально с log памятью, т.к. ей необходимо лишь дважды переписать вход, а затем добавить 2 ребра и 1 вершину.

f определяет корректную сводимость, т.к.

$(G; s; t) \in$ DPATH $\Leftrightarrow$ в G есть путь из s в t $\Rightarrow$

$\Rightarrow$ в G' есть пути $s \to t$ и $s' \to t'$, а также $s \to v$, $s' \to v$ $\Rightarrow$

$\Rightarrow$ s и s' слабо связаны

~~(G; s; t) ∈ DPATH ⇔ в G есть путь из s в t~~

$(G; s; s') \in$ WEAK-PATH $\Leftrightarrow \exists u$: есть пути $s \to u$ и $s' \to u$

ясно также, что $u = v$ (иначе u лежит в одной из копий графа G, но от одной из них нельзя добраться до второй) $\Rightarrow$ есть пути $s \to v$ и $s' \to v$ $\Rightarrow$

$\Rightarrow$ есть пути $s' \to t'$ и $s \to t$ $\Rightarrow$ $(G; s; t) \in$ DPATH

$\Downarrow$

WEAK-PATH $\in$ NL-hard

WEAK-PATH $\in$ NL, т.к. для него есть сертификат: он имеет вид $\underbrace{v}_{\text{вершина}} \mid \underbrace{s \to v \mid t \to v}_{\text{пути}}$. Его можно прочесть 1 раз, т.к. можно запомнить вершину v, а затем читать пути, и проверять, что они корректны, затратив log. память. Если пути корректны (начинаются в s и t соответственно, заканчиваются в v, проходят только по существующим ребрам), то s и v слабо связаны. ~~Ес~~

Если s и t слабо связаны, то найдется v, а значит и корректный сертификат.

$\Downarrow$

WEAK-PATH $\in$ NL; WEAK-PATH $\in$ NL-complete ч.т.д.

N1

пусть $f_1(n) = n^{33}$; $f_2(n) = n^{34}$

тогда $DTIME(n^2)/f_1(n) \subset DTIME(n^2)/f_2(n)$, т.к. если есть подсказка длины n^{33}, то есть и подсказка длины n^{34}, равная подсказке длины n^{33} с $n^{34} - n^{33}$ бесполезными символами в конце.

$DTIME(n^2)/f_2(n) \subset DTIME(n^2)/f_1(n)$, т.к. МТ работает за $O(n^2)$, а значит просто не успеет прочесть n^4; n^3 символов подсказки, а успеет прочесть лишь первые $O(n^2)$, а значит имеет смысл лишь подсказка длины $O(n^2)$, а от любой подсказки большей длины будут использованы лишь первые $O(n^2)$ символов.
при этом f_1 и f_2 не равны асимптотически, хотя
$DTIME(n^2)/f_1(n) = DTIME(n^2)/f_2(n)$

N3

$X_1 = BPP$, т.к. можно поменять принимающее и отвергающее состояния машины и тогда

X_1	$M'(x;r)=1$	$M'(x;r)=0$
$\in$	$>4/5$	$<1/5$
$\notin$	$<1/5$	$>4/5$

$\frac{4}{5} > \frac{2}{3} \Rightarrow X_1 = BPP$

BPP	1	0
$\in$	$\geq 2/3$	$\leq 1/3$
$\notin$	$\leq 1/3$	$\geq 2/3$

$X_2 = PP$, т.к. можно провести амплификацию

...

PP	1	0
$\in$	$>1/2$	$<1/2$
$\notin$	$<1/2$	$>1/2$

№2

построим алгоритм, определяющий принадлежность G-INDEX, с полиномиальной памятью.

вход: $(G; s; t; k)$

любой путь $s \to t$ в G имеет вид $su_1u_2\dots u_m t$, где все u_i различны. Пусть тогда алгоритм перебирает все последовательности вершин $u_1 u_2 \dots u_k$, где длина этой последовательности от 0 до $n-2$, ~~$u_i \in G$~~ $u_i \in V(G) \setminus \{s; t\}$, все u_i различны. Иными словами нужно перебрать все перестановки на $0; 1; \dots; n-2$ элементах. Это можно сделать с полиномиальной памятью, т.к. в каждый момент нужно хранить лишь текущую перестановку ~~д~~ длины $\leq n-2$, а получать следующую можно полиномиальным алгоритмом, а следовательно с полиномиальной памятью.

Для каждой перестановки нужно проверить за линейное время ($\Rightarrow$ полином. память), ~~есть~~ есть ли в G соответствующий путь, и если есть, вычесть из k единицу.

Если после перебора всех перестановок $k = 0$, то $(G; s; t; k) \in$ G-INDEX, иначе $(G; s; t; k) \notin$ G-INDEX.

Этот алгоритм доказывает, что G-INDEX $\in$ PSPACE

ч.т.д.

№6.

1) рассмотрим алгоритм, который выбирает случайную строку в качестве σ равновероятно.

пусть $S = |S_\sigma|$; $S_i = \begin{cases} 1, & i \in S_\sigma \\ 0, & \text{иначе} \end{cases}$; $S = \sum_{i=1}^{n} S_i$

$$ES = \sum_{i=1}^{n} ES_i$$

$$P(1 \notin S_\sigma) \leq \frac{1}{2^{\log n + 1}} = \frac{1}{2n} \Rightarrow P(1 \in S_\sigma) \geq 1 - \frac{1}{2n} = \frac{2n-1}{2n}$$

событие происходит, тогда и только тогда, когда ни одна из покрашенных единиц не попала в S_σ, единиц $\geq \log n + 1$, вероятность каждого попадания $\frac{1}{2}$

$P(i \in S_\sigma) \geq \frac{2n-1}{2n}$, где $i \in \{1; \dots; n\}$ аналогично

$ES_i = P(i \in S_\sigma) \cdot 1 + P(i \notin S_\sigma) \cdot 0 \geq \frac{2n-1}{2n}$

$ES \geq \frac{2n-1}{2n} \cdot n = \frac{2n-1}{2} = n - \frac{1}{2}$

но $|S_\sigma|$ целая $\Rightarrow$ $E|S_\sigma|_{|S_\sigma| \leq n} = n$ $\Rightarrow$ $\exists \sigma : S_\sigma = \{1; \dots ; n\}$

2) Дерандомизируем алгоритм методом условных мат. ожиданий ~~На каждом~~ будем строить σ по букве, на каждом шаге выбирая ту букву, при которой мат. ожидание S больше. ES можно считать за полином, т.к. для его подсчета нам хватит двух проходов по строке таблицы. Первый – если добавить к строке R; второй – если B.

Получившийся алгоритм полиномиален, т.к. работает за ~~[illegible]~~. ~~[illegible]~~ т.ч. каждой строке делает 2 прохода, во время каждого делает полином действий ~~[illegible]~~

~~Пересчет мат. ожидания производится так: [illegible] будет мощность S_σ после добавления~~

Пересчет мат. ожидания производится так:

мат. ожидание при добавлении R: $|S_\sigma^R| + \sum_{i \in \{1;2;\dots;n\} \setminus S_\sigma} P(i \in S_\sigma)$

при добавлении B: $|S_\sigma^B| + \sum_{i \in \{1;2;\dots;n\} \setminus S_\sigma} P(i \in S_\sigma)$

мощность уже набранного множества после добавления буквы R или B соответственно

каждый из этих пересчетов полиномиален

№4

$L \in RP^+ \iff \exists M$ - полиномиальная ВМТ:

1) $x \in L \Rightarrow \geq \frac{1}{2}$ путей заканчиваются в "принять"

2) $x \notin L \Rightarrow$ все заканчиваются в "отвергнуть"

при этом M - ПВМТ $\Rightarrow$ $z = p(|x|)$, M останавливается за $\leq p(|x|)$ шагов на x.

$\Downarrow$

RP^+	1	0
$\in$	$\geq 1/2$	$\leq 1/2$
$\notin$	0	1

$\Rightarrow RP^+ = RP$

CLIQUE = {(G; k) в G есть клика на k вершинах}

Пусть машине дают достаточно для того чтобы она могла воспринимать их как n ~~клик~~ подмножеств вершин размера k

Тогда она может для каждой из них проверить, клика ли это подмножество, за полином.

Если хоть одно из проверенных ~~клик~~ подмножеств клика, то вернуть 1, иначе — 0.

Доля неправильных ответов при $x \in L$ $\leq 1/2$ } доля неправильных при $x \notin L$ равна 0

$\Downarrow$

доля правильных при $x \in L$ $\geq 1/2$

$\Downarrow$

CLIQUE $\in RP^+$ ч.т.д.