Устойчивость супружеских пар и другие комбинаторные задачи

Введение в математический анализ алгоритмов

Дональд Э. Кнут

Содержание

От переводчиков русского издания

Лекции Дональда Кнута были впервые изданы в 1976 году, однако ничуть не потеряли своей актуальности. Идея и замысел этой книги ярко описаны автором в предисловии. Семь лекций не могут заменить знаменитых томов Д. Кнута "Искусство программирования", но, по нашему мнению, будут их удачным дополнением.

Мы старались сохранить оригинальный авторский стиль Кнута, благодаря которому его книги доставляют читателю не только радость познания, но и эстетическое наслаждение. В текст перевода добавлены ссылки на некоторые более поздние результаты по данной тематике.

Мы благодарны Дональду Кнуту за внимание к нашим замечаниям, Даниилу Мусатову за тщательную корректуру первоначального текста и Лёше Савватееву за поддержку при подготовке перевода.

Эдуард Лернер, Ольга Кашина

От переводчика английского издания

В известной игре "испорченный телефон" участники шёпотом по цепочке передают друг другу некоторую фразу. Последний из участников произносит услышанное им вслух, что обычно вызывает всеобщее веселье, поскольку окончательный вариант имеет мало общего с первоначальным. С лекциями профессора Д. Кнута происходили чуть более сложные метаморфозы — сначала он "прошептал" свои лекции студентам (их имена упомянуты в предисловии) на английском, а те, в свою очередь, "прошептали" их мне на французском. Теперь же я оглашаю их в обратном переводе. Надеюсь, что результат не вызовет того же эффекта, что происходит на вечеринке.

Говоря точнее, эта книга есть перевод исправленного и переработанного варианта монографии *Mariages stables et leurs relations avec d'autres problèmes combinatoires*, опубликованной в 1976 году издательством *Les Presses de l'Université de Montréal*. Основой её являются лекции, прочитанные профессором Д. Кнутом в Центре математических исследований (Centre de recherches mathématiques). Чтобы сохранить французский дух, мы оставили оригинальные французские имена[1], а также некоторые термины.

Хочу поблагодарить Луизу Летандр (Louise Letendre) за великолепный набор, Андре Монпети (André Montpetit) за техническую помощь в создании иллюстраций и в форматировании текста и автора за аккуратное редактирование перевода. Наконец, хочу выразить благодарность всем коллегам из Монреальского университета (Université de Montréal), которые охотно отвечали на мои вопросы о тонкостях перевода с французского на английский.

Мартин Гольдштейн (Martin Goldstein)

[1] В русском варианте они несколько изменены — замечание переводчиков.

Предисловие к первому изданию (на французском языке)

В течение последнего десятилетия наблюдается быстрое развитие информационных технологий. Что же касается дисциплины, предметом которой являются не информация, и не компьютеры как таковые, а процессы обработки информации, или алгоритмы, — то для неё наиболее подходящим, на мой взгляд, является название “Алгоритмика”. Вне зависимости от названия эта область знаний является весьма привлекательной по целому ряду причин.

Цель данной работы состоит в том, чтобы ознакомить читателя с основами анализа алгоритмов, причём сделать это с помощью примеров, а не систематического изложения теории. Надеюсь, что такой подход позволит читателю быстро войти в курс дела, познакомиться с идеями, используемыми в этой области, а также понять взаимосвязь анализа алгоритмов с другими математическими дисциплинами. Задача об устойчивых супружеских парах наилучшим образом соответствует этим целям: во-первых, её изучение не требует никаких предварительных знаний по алгоритмике, а во-вторых, она позволяет наглядно продемонстрировать основные методы анализа алгоритмов. Эта задача показывает, насколько интересным может быть анализ алгоритмов сам по себе, не говоря о его практической значимости.

Элементарный уровень изложения не предполагает от читателя опыта ни в анализе алгоритмов, ни в супружестве.

Хочу выразить благодарность доктору Андре Айзенштадту (André Aisenstadt), благодаря которому состоялась моя командировка в Монреаль, где я прочел студентам Монреальского университета цикл лекций, лёгших в основу этой брошюры. Общение с доктором Айзенштадтом доставило мне огромное удовольствие. Хочу также поблагодарить профессора Анатоля Жоффа (Anatole Joffe) за приглашение и оказанное мне и моей семье гостеприимство.

К сожалению, я не владел французским в совершенстве. Однако именно благодаря своим пробелам в знаниях французского я имел удовольствие работать с Паскалем Руссо (Pascale Rousseau) и Франсе Трошю (Français Trochu), тщательно переведшими на французский оригинальный английский тест этих лекций. Я также хотел бы поблагодарить одного из моих студентов — Бернара Мон-Рено (Bernard Mont-Reynaud) — за помощь в подготовке окончательной версии, и Мишлин Марано (Micheline Marano) и Жоанну Марку (Johanne Marcoux) за безошибочный набор текста.

Дональд Кнут, Стэнфорд, Калифорния, май 1976

N.B. При подготовке к печати третьего издания этой книги мной были исправлены некоторые опечатки, закравшиеся в первые два издания. Замечу, что некоторые из сформулированных здесь проблем по-прежнему остаются нерешёнными, особенно я ожидаю решения проблемы 2!

Д.Э.К., февраль 1995

ЛЕКЦИЯ 1

Введение, определения, примеры

Пусть H и F — два множества из n элементов. H есть *множество мужчин* $A_1, A_2, \ldots, A_n$; F есть *множество женщин* $a_1, a_2, \ldots, a_n$.[2] *Паросочетание* — это биекция из H в F, то есть множество из n моногамных разнополых браков. Далее множества H и F равноправны, и тот факт, что женщины обозначены строчными буквами, не имеет особого значения.

Предположим, что у каждого мужчины есть список женщин, упорядоченный по убыванию их "привлекательности" для него. Аналогично, у каждой женщины есть упорядоченный по тому же принципу список мужчин. Таким образом, предпочтения полов представлены двумя таблицами (матрицами) из n^2 элементов.

Паросочетание называется неустойчивым, если некие мужчина A и женщина a, не состоящие в браке между собой, оба предпочитают друг друга своим супругам. Такая "опасность измены" возникает, когда

- A женат на b;
- a замужем за B;
- A предпочитает a женщине b;
- a предпочитает A мужчине B.

(Мнение женщины b и мужчины B здесь не учитываются.) В дальнейшем такую ситуацию будем называть *адюльтером* Aa. Паросочетание без адюльтеров назовём *устойчивым.*

Пример 1. Четверо мужчин (A, B, C, D) состоят в браке с четырьмя женщинами (a, b, c, d).

Мужчины	**Порядок предпочтений**	**Женщины**	**Порядок предпочтений**
Anatole	$c \quad b \quad d \quad a$	alice	$A \quad B \quad D \quad C$
Bob	$b \quad a \quad c \quad d$	brigitte	$C \quad A \quad D \quad B$
Charles	$b \quad d \quad a \quad c$	carole	$C \quad B \quad D \quad A$
Dave	$c \quad a \quad d \quad b$	diana	$B \quad A \quad C \quad D$

Заметим, что эти предпочтения остаются неизменны.

Паросочетание $(\boldsymbol{A}a, B\boldsymbol{b}, Cc, Dd)$ неустойчиво, т.к. A и b предпочитают друг друга своим партнёрам (на что указывает жирный шрифт в обозначениях A и b).

[2] Мы сохранили оригинальные обозначения H и F, происходящие, соответственно, от французских слов *homme* (мужчина) и *femme* (женщина) — замечание переводчиков.

Предположим, что Aa и Bb развелись, и A женился на b. Тогда единственное, что остаётся a и B — заключить брак. В результате получим паросочетание $(A\boldsymbol{b}, Ba, \boldsymbol{C}c, Dd)$, которое также неустойчиво из-за предпочтений b и C.

Продолжая в том же духе, мы получим последовательность, завершающуюся устойчивым паросочетанием:

$\boldsymbol{A}a$	$B\boldsymbol{b}$	Cc	Dd	неустойчивое
$A\boldsymbol{b}$	Ba	$\boldsymbol{C}c$	Dd	неустойчивое
$A\boldsymbol{c}$	Ba	Cb	$\boldsymbol{D}d$	неустойчивое
Ad	Ba	Cb	Dc	♡ устойчивое ♡

Чтобы убедиться, что последнее паросочетание устойчиво, достаточно для каждого мужчины составить список женщин, предпочитаемых им своей супруге:

A предпочитает c и b;
B предпочитает b;
для C и D таких женщин нет.

Аналогичные списки предпочтений женщин (по отношению к своим супругам) имеют вид:

a предпочитает A;
c предпочитает C и B;
d предпочитает B;
для b таких мужчин нет.

Мужчина A не может улучшить свой выбор, так как ни c, ни b не предпочитают его своим мужьям. Мужчина B также не может улучшить свой выбор, так как b уже замужем за наилучшим для неё мужчиной. Поэтому полученное паросочетание устойчиво.

Мы получили бы тот же результат, рассматривая ситуацию с точки зрения женщин, то есть пытаясь улучшить выбор, сделанный a, c или d. Мы видим, что мужчины, которых они предпочитают своим мужьям, увы, не разделяют их чувств.

Серия разводов по вышеизложенным правилам не обязательно приведёт к устойчивому паросочетанию. Покажем это на следующем примере.

Пример 2.

Выбор мужчин		**Выбор женщин**	
A:	$b \quad a \quad c$	a:	$A \quad C \quad B$
B:	любой порядок	b:	$C \quad A \quad B$
C:	$a \quad b \quad c$	c:	любой порядок

$$
\begin{array}{ccc}
\boldsymbol{A}a & B\boldsymbol{b} & Cc \\
A\boldsymbol{b} & Ba & \boldsymbol{C}c \\
Ac & B\boldsymbol{a} & \boldsymbol{C}b \\
\boldsymbol{A}c & Bb & C\boldsymbol{a} \\
\boldsymbol{A}a & B\boldsymbol{b} & Cc
\end{array}
$$

Серия разводов и повторных браков вместо устойчивого решения приводит к зацикливанию. Тем не менее, устойчивые решения существуют, а именно, (Ab, Bc, Ca) и (Aa, Bc, Cb).

Пример 3 (Выбор в условиях циклических перестановок списков предпочтений).

Выбор мужчин					
A:	a	b	c	d	e
B:	b	c	d	e	a
C:	c	d	e	a	b
D:	d	e	a	b	c
E:	e	a	b	c	d

Выбор женщин					
a:	B	C	D	E	A
b:	C	D	E	A	B
c:	D	E	A	B	C
d:	E	A	B	C	D
e:	A	B	C	D	E

В этой задаче существует пять устойчивых паросочетаний:

$$
\begin{array}{ccccc}
Aa & Bb & Cc & Dd & Ee \\
Ab & Bc & Cd & De & Ea \\
Ac & Bd & Ce & Da & Eb \\
Ad & Be & Ca & Db & Ec \\
Ae & Ba & Cb & Dc & Ed
\end{array}
$$

В первом устойчивом решении каждый мужчина женат на самой желанной им женщине, в то время как каждая женщина замужем за самым нежеланным мужчиной. Второе (и каждое последующее) устойчивое решение изменяет ситуацию каждого ровно на одну градацию: улучшает для женщины и ухудшает для мужчины. Последнее решение является наилучшим для каждой женщины и наихудшим для каждого мужчины. В связи с таким конфликтом интересов возникает вопрос о критерии выбора единственного устойчивого решения. Например, мы можем придерживаться точки зрения женщин или же мужчин.

Покажем, что здесь существует только пять устойчивых решений. A должен жениться на a, b, c, d или e. Предположим, что устойчивое паросочетание содержит Aa. Тогда оно должно содержать Ee. В самом деле, если E был бы женат на b, c или d, то согласно таблице предпочтений E предпочитал бы a своей супруге, в то время, как a предпочитала бы его своему супругу A, а значит, паросочетание было бы неустойчивым. Таким образом, имеем

$$Aa \Rightarrow Ee.$$

Продолжая в том же духе, получаем

$$Aa \Rightarrow Ee \Rightarrow Dd \Rightarrow Cc \Rightarrow Bb \Rightarrow Aa$$

Чтобы устойчивое решение содержало Ab, оно должно содержать Ee или Ea, но как мы только что видели, $Ee \Rightarrow Aa$. Таким образом, получаем последовательность импликаций

$$Ab \Rightarrow Ea \Rightarrow De \Rightarrow Cd \Rightarrow Bc \Rightarrow Ab.$$

Аналогично, если устойчивое решение содержит Ac, то имеем

$$Ac \Rightarrow Eb \Rightarrow \text{ и т.д.}$$

Мы рассмотрели пример, когда существует n решений. Возникает вопрос: всегда ли существует n различных решений? Позднее (в лекции 2) мы докажем существование по крайней мере одного устойчивого решения. Можно подумать, что число устойчивых решений не превосходит $2n$ или же n^2, однако, как показывает следующий пример, их может быть много больше.

Пример 4 (n чётно).

Выбор мужчин				**Выбор женщин**		
1:	$\boxed{1}$	2	...	1:	...	1
2:	$\boxed{2}$	1	...	2:	...	2
3:	3	$\boxed{4}$	...	3:	...	3
4:	4	$\boxed{3}$	...	4:	...	4
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
$n-1$:	$\boxed{n-1}$	n	...	$n-1$:	...	$n-1$
n:	$\boxed{n}$	$n-1$	...	n:	...	n

Разобьем всех мужчин на “двойки”: первый-второй, третий-четвёртый и т.д. Предположим, что мужчины каждой “двойки” выбирают невест согласованно, а именно, их невесты имеют один и тот же рейтинг: либо первый, либо второй. Такой принцип отражён на приведённой выше схеме — для каждого мужчины номер выбранной им женщины заключен в рамочку. Можно построить $2^{n/2}$ различных паросочетаний.

Каждое полученное таким образом паросочетание является устойчивым. Действительно, пусть двое мужчин женятся на женщинах, которые стоят на втором месте в их списке предпочтений. Тогда измена с женщинами, занимающими первое место, невозможна, поскольку в рейтинге каждой из них соответствующий мужчина занимает последнее место.

Отсюда видно, что поиск всех устойчивых решений ведет к перебору экспоненциально растущего числа вариантов, что при достаточно больших значениях n может оказаться слишком трудоёмким.

Обобщение: задача о приёме n абитуриентов в m университетов

Пусть n абитуриентов поступают в m университетов, $n, m \geq 1$. Предположим, что университет с номером k может принять n_k абитуриентов. Без ограничения общности считаем, что $n_1 + n_2 + \ldots + n_m = n$. (Если общее число абитуриентов превышает суммарный план приёма в университеты, то вводим в рассмотрение фиктивный университет, считая, что его план приёма равен недостающему количеству мест и что этот университет является наименее предпочитаемым каждым из абитуриентов. Если же суммарный план приёма превышает общее число абитуриентов, то создаём нужное количество фиктивных студентов, наименее желательных для всех вузов. Тем самым получаем равенство $n_1 + n_2 + \ldots + n_m = n$.)

Каждый абитуриент имеет свой рейтинг университетов, каждый университет — свой рейтинг абитуриентов. Будем говорить об *устойчивом зачислении*, если никакой абитуриент A, не зачисленный в университет a, но предпочитающий a своему вузу, не окажется предпочтительнее для университета a, чем кто-то из его студентов.

Проблема сводится к задаче об устойчивых паросочетаниях заменой k-го университета на n_k его “клонов” с одинаковыми рейтингами абитуриентов. В списках предпочтений абитуриентов каждый университет заменяется на подписок его произвольно упорядоченных “клонов”. На практике задачи об устойчивых паросочетаниях решалась (на компьютере) для поиска устойчивого распределения абитуриентов по университетам Ирана и распределения интернов по американским больницам.

Обобщение: случай неполных списков предпочтений

Рассмотрим задачу об устойчивых паросочетаниях в случае, когда в рейтингах, составленных как мужчинами, так и женщинами, могут участвовать не все представители противоположного пола.

Будем считать, кроме того, что каждый человек вступает в брак только с тем, кто указан в списке его (её) предпочтений – он (она) скорее останется холостым (незамужней), чем заключит брак с тем, кого в этом списке нет.

Пусть, например, имеются следующие неполные списки:

$$\begin{array}{llll} A: & a & & \\ B: & c & a & b \\ C: & c & a & \end{array} \qquad \begin{array}{llll} a: & C & A & B \\ b: & B & A & C \\ c: & A & B & C \end{array}$$

Единственное возможное паросочетание здесь – $(Aa, \boldsymbol{B}b, C\boldsymbol{c})$; оно неустойчиво из-за B и c.

Таким образом, теорема существования устойчивых паросочетаний, справедливая для полных списков предпочтений, не распространяется на случай неполных списков.

Введём следующие обозначения.[3]

$$
\begin{aligned}
a \in A \quad &\Longleftrightarrow \quad \text{женщина } a \text{ есть в списке предпочтений мужчины } A\\
A \in a \quad &\Longleftrightarrow \quad \text{мужчина } A \text{ есть в списке предпочтений женщины } a\\
aAb \quad &\Longleftrightarrow \quad \text{для мужчины } A \text{ женщина } a \text{ лучше, чем } b\text{, или же } b \notin A\\
AaB \quad &\Longleftrightarrow \quad \text{для женщины } a \text{ мужчина } A \text{ лучше, чем } B\text{, или же } B \notin a
\end{aligned}
$$

Таким образом, паросочетание $(A_1a_1, A_2a_2, \ldots, A_na_n)$ устойчиво, тогда и только тогда, когда

(i) $a_k \in A_k$ и $A_k \in a_k$ при $1 \le k \le n$;

(ii) условия $A_ja_kA_k$ и $a_kA_ja_j$ не выполняются одновременно ни при каких j и k.

Одним из подходов к решению задачи порядка n является сведение её к задаче порядка $n-1$. Легко показать, что устойчивое паросочетание, содержащее пару A_na_n, существует тогда и только тогда, когда $a_n \in A_n$, $A_n \in a_n$ и существует устойчивое паросочетание в задаче порядка $n-1$ (её участников будем обозначать буквами B и b с индексами), определённой следующими условиями (здесь $1 \le i, j, k \le n-1$):

$$
\begin{aligned}
b_i \in B_j \quad &\Longleftrightarrow \quad a_i \in A_j \text{ и неверно, что } A_ja_nA_n \text{ и } a_nA_ja_i \text{ одновременно;}\\
B_i \in b_j \quad &\Longleftrightarrow \quad A_i \in a_j \text{ и неверно, что } a_jA_na_n \text{ и } A_na_jA_i \text{ одновременно;}\\
b_iB_jb_k \quad &\Longleftrightarrow \quad a_iA_ja_k \text{ или } b_k \notin B_j;\\
B_ib_jB_k \quad &\Longleftrightarrow \quad A_ia_jA_k \text{ или } B_k \notin b_j
\end{aligned}
$$

(последние две формулы означают, что за счёт выбывших из списка предпочтений мужчин и женщин условия $b_iB_jb_k$ и $B_ib_jB_k$ справедливы в большем числе случаев, чем соответствующие условия $a_iA_ja_k$ и $A_ia_jA_k$).

Преобразование неполных списков в полные

Полный список предпочтений можно создать из неполного, добавив нового мужчину V (вдовца) и новую женщину v (вдову), которые будут представлять худшие варианты друг для друга. Мужчину V поставим на последнее место в списке предпочтений (возможно, неполном) каждой женщины a_k, после чего дополним этот список всеми остальными мужчинами в произвольном порядке.

Женщину v поставим на последнее место в списке предпочтений (возможно, неполном) каждого мужчины A_k, после чего дополним этот список всеми остальными женщинами в произвольном порядке.

Имеют место следующие утверждения:

[3] Обратим внимание, что если $a \notin A$, $b \notin A$, то всегда справедливо aAb, bAa. Аналогичное утверждение верно и относительно предпочтений женщин — замечание переводчиков.

Теорема 1. *Полная система имеет устойчивое паросочетание с парой Vv тогда и только тогда, когда существует устойчивое паросочетание для неполной системы.*

Теорема 2. *Если полная система имеет устойчивое паросочетание с парой Vv, то эта пара входит во все устойчивые паросочетания для этой системы.*

Доказательство теоремы 2 следует из результатов, изложенных в Лекции 2.

Таким образом, для того чтобы сделать вывод о существовании устойчивого паросочетания в неполной системе, достаточно получить одно устойчивое паросочетание для полной системы. Если в нём нет пары Vv, то в неполной системе нет устойчивых решений. Иначе, исключая из найденного устойчивого решения полной системы пару Vv, мы получим устойчивое решение для неполной системы.

Упражнения

1. Пусть задано множество мужчин $A_1, A_2, \dots, A_n$. Для каждого мужчины A_k найдём такую женщину a_k, что пара $A_k a_k$ входит хотя бы в одно устойчивое паросочетание и не существует устойчивого паросочетания, в котором супругой A_k была бы более желанная для него женщина. Женщину a_k будем называть *оптимальной* для A_k. Пусть все списки предпочтений являются полными. Используя (неустановленное пока) существование хотя бы одного устойчивого паросочетания, докажите, что

   (a) $a_j \neq a_k$ для всех $j \neq k$;

   (b) паросочетание $(A_1 a_1, A_2 a_2, \dots, A_n a_n)$ устойчиво.

2. Найдите все устойчивые паросочетания для списков предпочтений:

$$\begin{array}{llllll} A: & e & d & c & b & a \\ B: & a & e & d & c & b \\ C: & b & a & e & d & c \\ D: & c & b & a & e & d \\ E: & d & c & b & a & e \end{array} \qquad \begin{array}{llllll} a: & A & B & C & D & E \\ b: & B & C & D & E & A \\ c: & C & D & E & A & B \\ d: & D & E & A & B & C \\ e: & E & A & B & C & D \end{array}$$

3. (Хванг (J. S. Hwang)) Предположим, что матрица предпочтений мужчин представляет собой латинский квадрат (каждый столбец есть перестановка имён женщин). Покажите, что все паросочетания, определяемые столбцами латинского квадрата, устойчивы тогда и только тогда, когда матрица предпочтений женщин двойственна к латинскому квадрату (то есть женщина a занимает j-е место в рейтинге мужчины A тогда и только тогда, когда тот занимает место с номером $n+1-j$ в рейтинге женщины a, здесь a и A — любые женщины и мужчины).

ЛЕКЦИЯ 2

Существование устойчивого паросочетания: основной алгоритм

Опишем алгоритм, позволяющий найти одно устойчивое паросочетание. В дальнейшем будем называть его основным алгоритмом. Тем самым мы получим конструктивное доказательство существования по крайней мере одного решения задачи об устойчивых паросочетаниях. В первой лекции мы видели, что случайная последовательность разводов не всегда завершается устойчивым паросочетанием. В основном алгоритме разводы заменены последовательностью помолвок. Инициатива выбора принадлежит мужчинам: они по очереди, один за другим, делают предложения женщинам. Женщины либо принимают предложения, либо отвергают их в соответствии со своими предпочтениями. Как мы увидим, это всегда приводит к устойчивому решению.

В алгоритме используются 3 переменные: k, X и x, — и две константы: n и Ω.

n — число мужчин (равное числу женщин);
k — число сформированных (пробных) пар;
X — мужчина, делающий предложение;
x — женщина, которой делается предложение;
Ω — (очень нежелательный) воображаемый мужчина.

Описание алгоритма

$k \leftarrow 0$; {все женщины (временно) помолвлены с Ω}
while $k < n$ **do**
  begin $X \leftarrow (k+1)$-ый мужчина;
  while $X \neq \Omega$ **do**
    begin $x \leftarrow$ лучшая кандидатура из текущего списка предпочтений мужчины X;
    if x предпочитает мужчину X своему жениху **then**
      begin
      помолвка X и x;
      $X \leftarrow$ предыдущий жених женщины x
      end;
    if $X \neq \Omega$ **then**
      вычёркиваем x из текущего списка предпочтений мужчины X;
  end;
  $k \leftarrow k+1$
end;
празднуем n свадеб.

Для читателей, не знакомых с языками программирования типа ALGOL'а, сделаем некоторые пояснения. Оператор $k \leftarrow 0$ присваивает переменной k значение 0. Команды, расположенные между словами **begin** и **end**, составляют блок операторов. Команды, помещённые в блок после конструкции **while** $\langle \ldots \rangle$ **do**, выполняются, пока справедливы условия, указанные в $\langle \ldots \rangle$.

Пример 1. Для понимания изложенного алгоритма применим его к Примеру 1 из прошлой лекции.

$$\begin{array}{lllll} A: & c & b & d & a \\ B: & b & a & c & d \\ C: & b & d & a & c \\ D: & c & a & d & b \end{array} \qquad \begin{array}{llllll} a: & A & B & D & C & \Omega \\ b: & C & A & D & B & \Omega \\ c: & C & B & D & A & \Omega \\ d: & B & A & C & D & \Omega \end{array}$$

Обратите внимание, что мужчина Ω добавлен в конец списка предпочтений каждой женщины.

На начальном шаге алгоритма переменная k принимает значение 0, и каждая женщина считается помолвленной с Ω.

Первый проход по внешнему циклу. После того, как мы убедились, что $k < n$, переменная X принимает значение A. Затем, убеждаясь, что $X \neq \Omega$, выполняем присвоение

$x \leftarrow$ лучшая кандидатура из текущего списка предпочтений мужчины X,

то есть

$$x \leftarrow c.$$

Таким образом, на данный момент переменные k, X и x принимают следующие значения:

$$k = 0, \quad X = A, \quad x = c.$$

Так как женщина c предпочитает мужчину A своему жениху Ω, c и A объявляют помолвку. Таким образом, X принимает значение Ω (который прежде был женихом c). Так как $X = \Omega$, то команду

вычёркиваем x из текущего списка предпочтений мужчины X;

выполнять не будем. Вместо этого мы возвращаемся к проверке

while $X \neq \Omega$ **do**.

Так как $X = \Omega$, то мы выходим из цикла, начинающегося этим оператором. Значение индекса k увеличивается на единицу $(k \leftarrow k + 1)$

Второй проход по внешнему циклу. Поскольку $k < n$, мы выполняем команду

$$X \leftarrow (k+1) - \text{ый мужчина},$$

и, таким образом, X принимает значение B; мужчина B выходит на "брачный рынок". Мы убеждаемся, что $X \neq \Omega$, и выполняем команду

$x \leftarrow$ лучшая кандидатура из текущего списка предпочтений мужчины X,

то есть
$$x \leftarrow b.$$
Таким образом, имеем
$$k = 1, \quad X = B, \quad x = b.$$
Сделав предложение женщине b, мужчина B создаёт пару Bb. X становится равным Ω, и мы снова увеличиваем значение k.

Третий проход по циклу. Этот проход отличается от двух предыдущих. Мы убеждаемся, что $k = 2 < n$. Таким образом, $X \leftarrow C$. Наилучшей кандидатурой в текущем списке предпочтений мужчины C является женщина b. Поэтому $x \leftarrow b$ и
$$k = 2, \quad X = C, \quad x = b.$$
Но женщина b предпочитает мужчину C своему жениху B. Поэтому мы формируем пару Cb и переменная X принимает значение B (это бывший жених b). Убедившись, что $X \neq \Omega$, мы вычёркиваем x из списка предпочтений X (то есть b из списка предпочтений B).

Так как X имеет значение B, команда
$$\textbf{while } X \neq \Omega \textbf{ do}.$$
означает продолжение выполнения внутреннего цикла (в роли потенциального жениха вновь выступает B). Наилучшая среди оставшихся кандидатура для B есть a. Следовательно, $x \leftarrow a$ и
$$k = 2, \quad X = B, \quad x = a.$$
Женщина a предпочитает мужчину B своему (наипротивнейшему) жениху Ω. Поэтому создаётся пара Ba, и переменной X присваивается значение Ω (раньше он был женихом a). Мы снова выходим из цикла, начинающегося оператором **while** $X \neq \Omega$ **do**, и увеличиваем k.

Четвёртый проход по циклу. Мы убеждаемся, что $k = 3 < n$. Поэтому $X \leftarrow D$, и мы получаем
$$k = 3, \quad X = D, \quad x = c.$$
Женщина c предпочитает мужчину D своему партнёру A. Она заменяет его (на D), и мужчина A выходит на “брачный рынок” (становится новым значением переменной X). Женщина c вычёркивается из списка предпочтений мужчины A. Наилучшей кандидатурой для A становится b. Следовательно,
$$k = 3, \quad X = A, \quad x = b.$$
Но b уже помолвлена с C, которого она предпочитает претенденту A, поэтому она не меняет партнёра. Мисс b вычёркивается из списка предпочтений

мужчины A, которому снова предстоит сделать предложение первой кандидатуре в оставшемся списке своих предпочтений. Итак,

$$k = 3, \quad X = A, \quad x = d.$$

Мисс d принимает предложение мужчины A, так как предпочитает его нежеланному Ω. Итак, $X \leftarrow \Omega$, и мы выходим из внутреннего цикла, увеличивая k.

Завершение внешнего цикла. Получив значение $k = 4 = n$, мы выходим из внешнего цикла

$$\textbf{while } k < n \textbf{ do}.$$

Теперь мы можем отпраздновать четыре свадьбы, поскольку, как будет показано ниже, полученное паросочетание является устойчивым.

Замечание. Может возникнуть вопрос: почему в алгоритме проверка условия $X \neq \Omega$ выполняется дважды:

$$\textbf{while } X \neq \Omega \textbf{ do} \ldots$$

и

$$\textbf{if } X \neq \Omega \textbf{ then} \ldots$$

Использование оператора **goto** позволило бы избежать лишних проверок, однако некоторые специалисты в области computer science считают, что лучше обходиться без него.

Доказательство корректности алгоритма

Математически строгое исследование алгоритмов началось не так давно (в последние 5–10 лет).[4] Ранее алгоритмы разрабатывались, исходя из интуитивных представлений, а их работоспособность проверялась с помощью написанных на их основе компьютерных программ. Для того, чтобы доказать корректность алгоритма и оценить его эффективность, необходимо выяснить смысл всех переменных и их взаимосвязь на каждом шаге алгоритма. Доказательство проводится методом математической индукции по номеру итерации.

Сначала заметим следующие свойства, справедливые для любой итерации алгоритма.

Предложение 1. *Если женщина a вычёркивается из списка предпочтений мужчины A, то ни одно устойчивое паросочетание не будет содержать пары Aa.*

Доказательство. При выполнении операции “вычеркнуть женщину x из списка предпочтений мужчины X” в ходе реализации алгоритма при значениях переменных $x = a$ и $X = A$ возможны две ситуации:

[4] Напомним, что лекция 2 была впервые издана в 1976 году — замечание переводчиков.

(i) Мужчина A сделал предложение женщине a, но та предпочла ему своего жениха B.

(ii) Женщина a, помолвленная с A, оставляет его, получив более привлекательное предложение от B.

В обоих случаях B более предпочтителен для a, чем A, при этом для B кандидатура a предпочтительнее всех остальных женщин в его текущем списке. Докажем предложение 1 от противного. Рассмотрим сначала случай, когда текущий список предпочтений B совпадает с начальным. Предположим, что устойчивое паросочетание содержит пару Aa. Ввиду опасности адюльтера Ba наличие пары Aa означает, что B женат на ком-то, кто для него лучше, чем a. Однако в рейтинге мужчины B кандидатура a стоит на первом месте. Противоречие. Предполагая, что предложение 1 имеет место на всех предыдущих итерациях алгоритма, по индукции обобщаем этот результат на произвольный случай. □

Предложение 2. *Пусть в исходном списке предпочтений мужчины A кандидатура a предшествовала его нынешней невесте. Тогда a отвергла A ради кого-то другого.* □

Предложение 3. *Никакие две женщины не могут одновременно быть помолвлены с одним мужчиной, за исключением Ω.* □

Предложение 4. *Никакая женщина никогда не ухудшает свой выбор в ходе алгоритма.* □

Предложение 5. *Список предпочтений любого мужчины никогда не становится пустым.*

Доказательство. Предположим противное. В силу предложения 2 это означает, что этого мужчину отвергли все женщины. Но тогда в силу предложений 3 и 4 каждая из женщин имеет своего “бойфренда”, отличного от Ω. Таким образом, должно существовать n мужчин, кроме A. Противоречие. □

Заметим, что только нарушение предложения 5 могло бы привести к невозможности составить паросочетание с помощью описанного выше алгоритма, а так как предложение 5 доказано, паросочетание найдётся всегда.

Описанный алгоритм конечен, поскольку после каждого очередного предложения либо сокращается список предпочтений всех мужчин, либо значение переменной k увеличивается на единицу.

Предложение 6. *Полученное паросочетание устойчиво.*

Доказательство. Пусть A женат не на a, но предпочитает её своей супруге. Это означает, что a предпочла ему кого-то другого (предложения 2, 4), следовательно адюльтер Aa невозможен. □

На самом деле алгоритм вырабатывает *оптимальный вариант для каждого мужчины*. Каждый мужчина женат на лучшей из возможных кандидатур: не существует устойчивого паросочетания, в котором его супругой была бы более желанная для него женщина. (Это следует из предложения 1. Заметим, что мы не использовали предложение 1 в обосновании алгоритма — оно понадобилось нам лишь здесь, при доказательстве более сильного результата.)

Поскольку вырабатываемое алгоритмом решение является *оптимальным* для всех мужчин, их нумерация не имеет значения, хотя при работе алгоритма используется некоторая их очерёдность.

В то же время, полученный результат является *наихудшим для всех женщин.* (То есть партнёр, назначенный женщине алгоритмом, нравится ей не больше её партнёра в любом устойчивом паросочетании.)

Доказательство. Пусть в решении, найденном алгоритмом, содержится пара Aa (это означает в силу оптимальности найденного решения для всех мужчин, что a для A — наилучшая из всех реальных кандидатур). Предположим, что в другом устойчивом паросочетании содержатся пары Ba и Ab, причём a предпочитает A мужчине B. Тогда A должен предпочитать b кандидатуре a, что противоречит предположению об оптимальности a для A. □

С помощью того же алгоритма можно получить решение, наилучшее для всех женщин (и наихудшее для мужчин) — в этом случае предложения "руки и сердца" должны делать женщины.

Конфликт интересов

Тот факт, что найденное решение является "наилучшим для мужчин и наихудшим для женщин одновременно" представляет собой частный случай более общего результата.[5]

Теорема 3. *Если некоторое устойчивое паросочетание содержит пару Aa, а некоторое другое устойчивое паросочетание — пары Ab и Ba, то возможна одна из двух ситуаций:*

$$bAa \text{ и } AaB$$

или

$$aAb \text{ и } BaA.$$

(Иными словами, любое другое устойчивое паросочетание лучше для одного из супругов и хуже для другого.)

Доказательство. По определению устойчивости новые партнёры бывших супругов A и a не могут одновременно оказаться хуже для них обоих. Остаётся показать, что они не могут быть и лучше для A и a одновременно.

[5]Фактически, теорема 3 ниже — это утверждение об оптимальности по Парето множества устойчивых паросочетаний как группового соответствия, осуществляемого мужчинами и женщинами — примечание переводчиков.

Пусть $A = X_0, a = x_0, b = x_1$. Предположим, что bAa, то есть $x_1X_0x_0$. В первом устойчивом паросочетании X_0 имеет худшую (из двух) партию, следовательно, x_1 выбрала того, кто желаннее ей, чем X_0.

Пусть в первом устойчивом паросочетании x_1 замужем за X_1. Тогда $X_1x_1X_0$. Во втором решении x_1 имеет худшую (из двух) партию, значит, X_1 сделал лучший выбор. Пусть во втором устойчивом паросочетании X_1 женат на x_2. Тогда $x_2X_1x_1$ и так далее.

Получаем последовательность:

$X_0x_0,\quad X_1x_1,\quad X_2x_2,\quad \ldots$ в первом устойчивом паросочетании,

$X_0x_1,\quad X_1x_1,\quad X_2x_3,\quad \ldots$ во втором устойчивом паросочетании,

где

$$x_{k+1}X_kx_k \text{ и } X_{k+1}x_kX_k \text{ для всех } k \geq 0.$$

Поскольку число мужчин конечно, найдутся номера j и k такие что $j < k$ и $X_j = X_k$. Пусть j и k — наименьшие из таких номеров. Имеем $x_j = x_k$. Тогда $j = 0$, так как иначе во втором устойчивом паросочетании содержались бы пары $X_{k-1}x_k = X_{k-1}x_j$ и $X_{j-1}x_j$. (Но тогда $X_{j-1} = X_{k-1}$, что противоречит минимальности j.) Поэтому второе устойчивое паросочетание содержит $X_{k-1}x_0$. Но $x_0 = a$. Тогда по условию теоремы $X_{k-1} = B$. Учитывая, что $X_kx_kX_{k-1}$, получаем AaB. □

Следствие. *Для двух произвольных устойчивых паросочетаний I и II имеет место свойство: если для всех мужчин I не хуже, чем II, то для всех женщин I не лучше, чем II.*

Доказательство теоремы 2 из лекции 1

Напомним, что упомянутая выше теорема утверждает следующее. Пусть A и a представляют собой наихудшие кандидатуры друг для друга, и некоторое устойчивое паросочетание содержит пару Aa. Тогда эта пара содержится во *всех* устойчивых паросочетаниях.

Доказательство. Устойчивое паросочетание, наихудшее с точки зрения всех женщин, содержит пару Aa. Одновременно оно является наилучшим с точки зрения всех мужчин. Учитывая "симметричность" решений с точки зрения мужчин и женщин, получаем утверждение теоремы. □

Анализ алгоритма

Определим число шагов алгоритма. Проанализируем алгоритм с помощью блок-схемы на Рис. 1.

Число, написанное рядом со стрелкой, указывает количество выполнений соответствующей операции. Число мужчин, ра́вно как и женщин, есть заданная константа n. Обозначим через N общее число предложений, сделанных "соискателями", оно зависит от списков предпочтений.

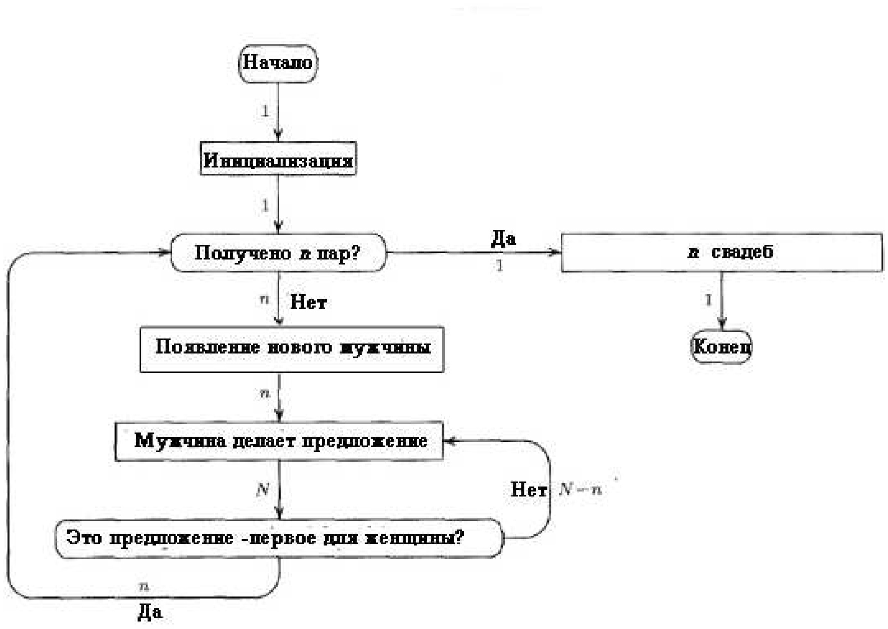


Рис. 1: Схема основного алгоритма.

Так как время выполнения алгоритма пропорционально N, нас интересует среднее значение N при случайном формировании списков предпочтений. Мы исследуем его в следующей лекции. Кроме того, интересно максимальное (то есть наихудшее с точки зрения трудоёмкости алгоритма) значение N. Мы изучим этот вопрос с помощью следующих упражнений.

Упражнения

1. Докажите, что согласно решению, найденному по описанному алгоритму, не более одного мужчины окажется помолвленным с наихудшей для него кандидатурой. (Следствие: если в некотором устойчивом паросочетании двое (или более) мужчин получают наихудший вариант невесты, то это означает, что существует, по крайней мере, два устойчивых паросочетания.)

2. Докажите, что результат, полученный в упражнении 1, является неулучшаемым: возможно, что один из мужчин получает наихудший вариант, а остальные $n-1$ человек — предпоследний вариант из своих списков предпочтений. (В этом случае общее число предложений равно $n+(n-1)(n-1)=n^2-n+1$.)

ЛЕКЦИЯ 3

Принцип отложенных решений: накопление купонов

В этой лекции мы оценим среднее число предложений, сделанных в процессе работы основного алгоритма.

Вначале рассмотрим следующую задачу, с помощью которой опишем один технический приём. Он потребуется нам далее.

Пасьянс "Циферблат"

Колоду из 52 карт тасуют, а затем карты раскладывают "рубашкой" вверх по кругу так, чтобы получилось подобие циферблата часов (см. Рис. 2). Каждый "час" соответствует номеру стопки (достоинству карты), за следующими исключениями:

- стопка на "1" называется "Туз"
- стопка на "11"называется "Валет"
- стопка на "12" называется "Дама"
- стопка в центре называется "Король".

Игра состоит в том, чтобы по очереди раскрывать карты согласно следующему правилу:

- первой раскрывается карта из стопки "Король";
- раскрытая карта помещается рядом с соответствующей стопкой (если, например, это будет карта "7", то ее следует положить около стопки "7");

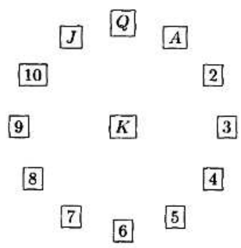


Рис. 2: Схема расположения стопок карт в пасьянсе "Циферблат".

- та же процедура повторяется с соответствующей стопкой (в нашем примере мы раскроем карту из стопки "7". Пусть, для определённости, достоинства карт совпадают с десятичными знаками дробной части числа Эйлера $e = 2,71828\ldots$. Тогда раскрытой картой будет "Туз", и мы поместим его около стопки "Туз", и так далее.);

Игра завершается, когда какая-то из тех стопок, откуда мы должны будем взять карту, окажется пустой. Это произойдёт, когда будет раскрыт последний король. Если при этом открыты будут все карты, то говорят, что пасьянс сошёлся.

Оказывается, можно узнать, сойдётся пасьянс или нет, не раскладывая его. Всё зависит от нижних карт всех стопок, расположенных по кругу. Пусть, например, эти карты таковы, как показано на Рис. 3.

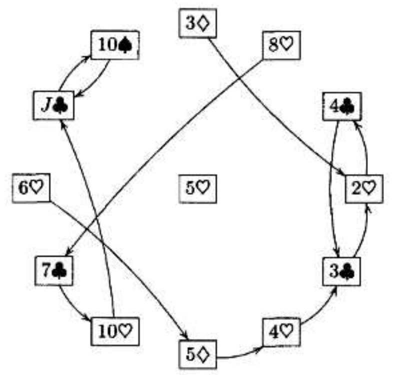


Рис. 3: Граф взаимосвязей нижних карт.

Стрелками указано новое местоположение каждой из карт. Стопка "Король" не имеет исходящей стрелки. Получаем *ориентированный граф* с 12 рёбрами и 13 вершинами.

Если в нём есть хотя бы один цикл, то пасьянс не сойдётся, поскольку игрок не сможет раскрыть все карты. Если же циклов нет, то пасьянс сойдётся независимо от расположения остальных 40 карт.[6]

Вероятность выигрыша составляет 1/13. Показать это можно двумя способами:

1. первый способ состоит в том, чтобы рассмотреть все возможные положения последних карт в стопках, расположенных по кругу, и подсчитать количество тех вариантов, в которых не образуется циклов.

[6]Аккуратное доказательство этих утверждений несложно — см. M.W.Ecker, "How to Win (or Cheat) in the Solitaire Game of "Clock"," Mathematics Magazine, Vol. 55, No. 1 (Jan., 1982), pp. 42-43. Впрочем, как можно убедиться из нижеизложенного, для оценки вероятности выигрыша достаточно использовать сделанное ранее замечание о том, что последним всегда открывается король — замечание переводчиков.

2. другой, более простой, метод использует *принцип отложенных решений*: “не делай сегодня то, что можно отложить на завтра”. В частности, достоинство карты должно определяться (случайным образом) не ранее того момента, когда она должна быть раскрыта.

Мы раскрываем карты согласно правилам игры. Как только четвёртый король будет раскрыт, оставшиеся карты мы открываем в некотором порядке. (Например, сначала открываем все карты стопки “Туз”, затем — стопки “2” и так далее.) Таким образом, в случайном порядке будут раскрыты все карты колоды. Для начального расположения карт существует 52! вариантов, столько же вариантов существует при раскрывании их в заданном нами порядке. Таким образом, легко подсчитать число выигрышных вариантов — это те случаи, когда последняя карта является королём.

Оценка среднего числа предложений

Найдём оценку среднего числа предложений, сделанных потенциальными женихами согласно алгоритму поиска устойчивого паросочетания. Предположим, что матрица предпочтений женщин постоянна, а матрица предпочтений мужчин формируется случайным образом. Для каждого мужчины существует $n!$ перестановок женщин, то есть $(n!)^n$ равновероятных матриц предпочтений. Оценим среднее число предложений (итераций) в алгоритме.

Будем использовать принцип отложенных решений. Каждый мужчина не формирует заранее своего списка предпочтений (выбирая его из $n!$ вариантов), а принимает решение в последний момент. Иными словами, каждый раз, когда наступает его очередь делать предложение, он адресует его случайно выбранной женщине. В пасьянсе это соответствует случайному выбору карты из числа оставшихся после окончания игры.

Однако оценить среднее число предложений $\mathrm{E}(N)$ всё ещё трудно. Необходимо упростить задачу.

Дополнительные предположения, упрощающие задачу

Предположим, что все мужчины страдают *“амнезией”*: они не помнят тех женщин, кому уже делали предложение.

Это эквивалентно перемешиванию всей колоды перед выбором из неё очередной карты. Порядок, в котором эти карты появляются *впервые*, задаёт случайную перестановку.

Среднее значение числа предложений, сделанных забывчивыми мужчинами, даёт верхнюю оценку для искомой величины $\mathrm{E}(N)$.

Действительно, в основном алгоритме количество сформированных пар увеличивается на единицу каждый раз, когда какой-либо женщине впервые делается предложение (согласно алгоритму она принимает первое предложение). Алгоритм завершает работу, как только каждой женщине хотя бы раз было сделано предложение. В случае с забывчивыми мужчинами алгоритм вырабатывает случайную последовательность женщин и заканчивает работу, как только в неё вошли все женщины (хотя бы по одному разу).

Рассмотрим следующий пример. Пусть мужчины A, B, C, D имеют следующий рейтинг у женщин a, b, c, d:

$$\begin{array}{lllll} a: & C & A & D & B \\ b: & B & D & A & C \\ c: & C & A & B & D \\ d: & C & D & B & A \end{array}$$

Рассмотрим последовательность предложений

$$d, b, d, b, c, c, b, c, d, a,$$

сделанных мужчинами. Согласно алгоритму предложения делались в следующем порядке:

A делает предложение $d \Rightarrow$ формируется пробная пара Ad

B делает предложение $b \Rightarrow$ формируется пробная пара Bb

C делает предложение $d \Rightarrow$ формируется пробная пара Cd.

Поскольку d предпочитает C своему жениху A, то A становится новым кавалером. A делает предложение b, но та его отвергает, поскольку предпочитает ему B.

A делает предложение $c \Rightarrow$ формируется пробная пара Ac.

Таким образом, получаем следующий список неполных предпочтений:

$$\begin{array}{llll} A: & d & b & c \\ B: & b & & \\ C: & d & & \\ D: & & & \end{array}$$

Этот процесс продолжается до тех пор, пока не будет сделано предложение последней из женщин, впервые появившейся в приведённой выше последовательности (то есть a).

D делает предложение c; она его отвергает

D делает предложение b; она его отвергает

D делает предложение c; она его снова отвергает.

Это первое “избыточное” предложение (по сравнению с основным алгоритмом).

D делает предложение d; она его отвергает

D делает предложение $a \Rightarrow$ формируется пробная пара Da.

Список неполных предпочтений принимает следующий вид:

$$\begin{array}{lllll} A: & d & b & c & \\ B: & b & & & \\ C: & d & & & \\ D: & c & b & d & a \end{array}$$

Таким образом сделано 10 предложений, одно из которых избыточно.

Заметим, что для получения устойчивого паросочетания с помощью основного алгоритма не требуется, вообще говоря, просматривать все списки предпочтений, потому мы и получили неполные списки.

Длина последовательности, необходимой для получения устойчивого паросочетания с помощью описанного алгоритма, равна числу предложений, сделанных в процессе его выполнения. Таким образом, средняя длина таких последовательностей равна *математическому ожиданию* $E(N)$ *числа предложений*, сделанных забывчивыми кавалерами. Любая последовательность, в которой каждая женщина появляется хотя бы однажды, задаёт устойчивое паросочетание. Исследование закона распределения длин таких последовательностей составляет *задачу собирателя купонов*.

Задача собирателя купонов

Пусть каждый раз при приобретении коробки стирального порошка покупатель получает купон. Всего существует n видов купонов. Выбор купона каждый раз происходит случайным образом. Сколько в среднем ему нужно купить коробок стирального порошка, чтобы собрать все n купонов?

Пример 1. Пусть $n = 10$. Нас интересует число шагов, необходимых для сбора всех цифр $0, 1, 2, \ldots, 9$ в десятичной записи чисел π и e.

$$\begin{array}{cccccccccccccccc} \mathbf{3} & \mathbf{1} & \mathbf{4} & 1 & \mathbf{5} & \mathbf{9} & \mathbf{2} & \mathbf{6} & 5 & 3 & 5 & \mathbf{8} & 9 & \mathbf{7} & 9 & 3 \\ & 2 & 3 & 8 & 4 & 6 & 2 & 6 & 4 & 3 & 3 & 8 & 3 & 2 & 7 & 9 \\ & 5 & \mathbf{0} & & & & & & & & & & & & & \\ \mathbf{2} & \mathbf{7} & \mathbf{1} & \mathbf{8} & 2 & 8 & 1 & 8 & 2 & 8 & \mathbf{4} & \mathbf{5} & \mathbf{9} & \mathbf{0} & 4 & 5 \\ & 2 & \mathbf{3} & 5 & 3 & \mathbf{6} & & & & & & & & & & \end{array}$$

Введём обозначения. Пусть у нас уже имеется m купонов, $m = 0, \ldots, n-1$. Обозначим через $p_k^{(m)}$ вероятность того, что для получения нового купона покупателю придётся приобрести ровно k, $k = 1, 2, \ldots$, коробок порошка (индекс m является фиксированным, мы будем иногда его опускать). Положим $q_k^{(m)} = \sum_{k=1}^{\infty} p_k^{(m)}$, для всех $k \leq 1$, то есть

$$\begin{array}{lcr} q_1 & = & p_1 + p_2 + p_3 + p_4 + \ldots \\ q_2 & = & p_2 + p_3 + p_4 + \ldots \\ q_3 & = & p_3 + p_4 + \ldots \\ \vdots & & \\ q_k & = & p_k + p_{k+1} + \ldots \\ \vdots & & \end{array}$$

Таким образом, $q_k^{(m)}$ есть вероятность того, что придётся купить как минимум k коробок. Имеем

$$q_1 = 1; \qquad p_k = q_k - q_{k+1}.$$

По определению среднее число купленных коробок, необходимое для получения нового купона, равно

$$p_1 + 2p_2 + 3p_3 + \ldots = q_1 + q_2 + q_3 + \ldots.$$

При покупке одной новой коробки вероятность получить купон, совпадающий с одним из m уже имеющихся, есть m/n. Отсюда

$$q_1^{(m)} = 1; \quad q_2^{(m)} = \frac{m}{n}; \quad q_3^{(m)} = \left(\frac{m}{n}\right)^2; \quad \ldots$$

$$q_1 + q_2 + q_3 + \ldots = 1 + \frac{m}{n} + \left(\frac{m}{n}\right)^2 + \ldots = \frac{n}{n-m}.$$

Среднее число коробок, которое придётся купить, чтобы собрать все n купонов, равно сумме вычисленных средних для $m = 0, 1, \ldots, n-1$, то есть

$$\begin{aligned} \frac{n}{n-0} + \frac{n}{n-1} + \frac{n}{n-2} + \ldots + \frac{n}{n-(n-1)} &= \\ &= n\left(1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}\right) = \\ &= nH_n, \end{aligned}$$

где H_n есть сумма n первых членов *гармонического ряда* $1 + 1/2 + 1/3 + \ldots$.

Выводы

Мы нашли *верхнюю оценку среднего числа предложений* $E(N)$, сделанных в процессе нахождения устойчивого паросочетания. Действительно, средняя длина последовательности женщин, где каждая из n кандидатур появляется хотя бы однажды, равна среднему числу коробок, которые нужно купить, чтобы собрать n купонов, то есть nH_n. Известно, что

$$H_n = \ln n + \gamma + \frac{1}{2n} - \frac{1}{12n^2} + \varepsilon,$$

где $0 < \varepsilon < 1/(120n^4)$, γ — константа Эйлера, $\ln$ обозначает натуральный логарифм. Таким образом, для достаточно больших n среднее число предложений примерно равно $n \ln n$.

Можно уточнить эту оценку, рассмотрев модель с *частичной амнезией*.

Частичная амнезия

Пусть, по-прежнему, мужчины имеют плохую память, но не настолько, чтоб делать предложение одной и той же женщине два раза подряд. Эта модель даст лучшую верхнюю оценку среднего числа предложений.

Пусть m женщин уже приняли сделанные им предложения, $m = 0, 1, \dots,$ $n-1$. Среднее число предложений, которые предстоит сделать для формирования очередной пары, можно вычислить по (слегка модифицированной) схеме собирателя купонов:

$$q_1 = 1; \quad q_2 = \frac{m}{n}; \quad q_3 = \frac{m}{n}\frac{m-1}{n-1}; \quad q_4 = \frac{m}{n}\left(\frac{m-1}{n-1}\right)^2; \quad \dots$$

$$\begin{aligned} q_1 + q_2 + q_3 + \dots &= 1 + \frac{m}{n}\left(1 + \frac{m-1}{n-1} + \left(\frac{m-1}{n-1}\right)^2 + \dots\right) = \\ &= \frac{n}{n-m} - \frac{m}{n(n-m)}. \end{aligned}$$

Вероятность того, что потребуется сделать хотя бы одно предложение (для формирования очередной пары), очевидно, равна единице. Вероятность того, что первое предложение будет сделано одной из m женщин, $q_2 = m/n$. Вероятность того, что второе предложение не приведёт к увеличению числа сформированных пар (то есть вероятность того, что выбранная женщина окажется одной из $m-1$ неподходящих кандидатур при общем числе вариантов равном $n-1$), равна $(m-1)/(n-1)$.

Тогда среднее число предложений $\mathrm{E}(N)$ легко вычислить по формуле

$$\begin{aligned} \mathrm{E}(N) &= \sum_{m=0}^{n-1} \frac{n^2 - m}{n(n-m)} = \sum_{k=1}^{n} \frac{n^2 - (n-k)}{nk} = \\ &= (n-1)\sum_{k=1}^{n}\frac{1}{k} + \frac{1}{n}\sum_{k=1}^{n} 1 = \\ &= (n-1)H_n + 1. \end{aligned}$$

Таким образом, справедлива следующая теорема.

Теорема 4. *Пусть матрица предпочтений женщин зафиксирована произвольно. Тогда среднее число предложений, сделанных в ходе выполнения алгоритма, не превосходит* $(n-1)H_n + 1$.

Иными словами, среднее число предложений, сделанных мужчиной, не превосходит $\ln n + \gamma$. Примерно то же составляет оценку сверху среднего числа предложений, получаемых каждой женщиной. Можно показать, что её окончательным выбором оказывается мужчина, занимающий в её рейтинге примерно $n/\ln n$-е место.

Упражнения

1. При заданной матрице предпочтений женщин

$$\begin{array}{llll} a: & A & B & C \\ b: & B & C & A \\ c: & C & A & B \end{array}$$

   покажите, что из $216 = (3!)^3$ возможных матриц предпочтений мужчин ровно

   48 матриц потребуют 3 итерации основного алгоритма (предложений),
   72 матрицы — 4 итерации,
   60 матриц — 5 итераций,
   30 матриц — 6 итераций,
   6 матриц — 7 итераций.

   (Таким образом, среднее число предложений равно $4\frac{5}{12}$; верхняя оценка равна $2H_3 + 1 = 4\frac{8}{12}$.)

2. Для заданной последовательности женщин (в порядке получения ими предложений от мужчин) *dbdbccbcda* найдите вероятность того, что "напрасно" будут сделаны k предложений, где $k = 0, 1, 2, \ldots$, при условии, что все мужчины страдают полной амнезией, а матрица предпочтений женщин сформирована случайным образом.

ЛЕКЦИЯ 4

Теоретические основы: применение в задаче о кратчайшем пути

В этой лекции мы рассмотрим следующие вопросы:

1. Краткий обзор теории дискретной вероятности.
2. Дисперсия в задаче собирателя купонов.
3. Анализ основного алгоритма поиска устойчивого паросочетания в наихудшем случае.
4. Применение к *задаче о кратчайшем пути в графе*.

1. Теория дискретной вероятности

Производящие функции

Пусть X — *дискретная случайная величина*, принимающая значения в множестве $\mathbb{N}$ неотрицательных целых чисел. Пусть p_k — вероятность того, что $X = k$; таким образом, $p_k \geq 0$ и $p_0 + p_1 + p_2 + \ldots = 1$.

Производящей функцией для этого ряда распределения случайной величины называется выражение

$$P(z) = p_0 + p_1 z + p_2 z^2 + \cdots = \sum_{k \geq 0} p_k z^k.$$

Заметим, что $P(1) = 1$.

Математическое ожидание $\mathrm{E}(X)$ и дисперсию $\mathrm{V}(X)$ случайной величины X можно вычислить с помощью значений производных производящей функции в точке $z = 1$. Действительно,

$$\begin{aligned}
P(z) &= p_o + p_1 z + p_2 z^2 + \ldots = \sum_{k \geq 0} p_k z^k; \\
P'(z) &= p_1 + 2p_2 z + 3p_3 z^2 + \ldots = \sum_{k \geq 0} k p_k z^{k-1}; \\
P''(z) &= 2p_2 + 6p_3 z + 12 p_4 z^2 + \ldots = \sum_{k \geq 0} k(k-1) p_k z^{k-2}.
\end{aligned}$$

Для математического ожидания $\mathrm{E}(X)$ имеем

$$\mathrm{E}(X) = \sum_{k \geq 0} k p_k = P'(1).$$

По определению дисперсии $\mathrm{V}(X) = \mathrm{E}(X - \mathrm{E}(X))^2$ имеем

$$\begin{aligned} \mathrm{V}(X) &= \sum_{k\geq 0}(k - \mathrm{E}(X))^2 p_k = \\ &= \sum_{k\geq 0} k^2 p_k - 2\mathrm{E}(X)\sum_{k\geq 0} kp_k + \mathrm{E}(X)^2 \sum_{k\geq 0} p_k = \\ &= \sum_{k\geq 0} k(k-1)p_k + \sum_{k\geq 0} kp_k - 2\mathrm{E}(X)^2 + \mathrm{E}(X)^2 = \\ &= P''(1) + P'(1) - P'(1)^2. \end{aligned}$$

Неравенство Чёбышева–Бьенамэ

Обозначим через $\sigma = \sqrt{\mathrm{V}(X)}$ среднее квадратичное отклонение случайной величины X с рядом распределения p_k, $k \geq 0$.

Для любого числа $m \geq 1$ вероятность того, что $|X - E(X)| > m\sigma$ не превышает $1/m^2$.

Это утверждение, называемое неравенством Чёбышева (оно также приписывается Бьенамэ), верно для любых случайных величин. Таким образом, как минимум в 99% случаев справедливы неравенства

$$\mathrm{E}(X) - 10\sigma \leq X \leq \mathrm{E}(X) + 10\sigma.$$

Доказательство. Обозначим через p вероятность того, что $|X - \mathrm{E}(X)| > m\sigma$. Тогда при $p > 0$ имеем

$$\begin{aligned} \mathrm{V}(X) &= \mathrm{E}((X - \mathrm{E}(X))^2) > \\ &> p(m\sigma)^2 + (1-p)\cdot 0 = \\ &= pm^2\mathrm{V}(X). \end{aligned}$$

Иными словами, $p < 1/m^2$. □

Независимые случайные величины

Пусть Y – ещё одна случайная величина, принимающая значение k с вероятностью q_k. Соответствующая производящая функция имеет вид

$$Q(z) = \sum_{k\geq 0} q_k z^k.$$

Случайные величины X и Y называются независимыми, если при всех j и k вероятность того, что $X = j$ и одновременно $Y = k$, равна $p_j q_k$.

Если случайные величины X и Y независимы, то вероятность того, что $X + Y = m$, задаётся формулой

$$r_m = p_0 q_m + p_1 q_{m-1} + \ldots + p_{m-1} q_1 + p_m q_0.$$

В этом случае производящая функция новой случайной величины $X + Y$ есть произведение производящих функций случайных величин X и Y. Действительно,

$$\begin{aligned}\sum_{m\geq 0} r_m z^m &= \sum_{m\geq 0}\Big(\sum_{\substack{j,k\geq 0\\ j+k=m}} p_j q_k\Big) z^m = \\ &= \Big(\sum_{j\geq 0} p_j z^j\Big)\Big(\sum_{k\geq 0} q_k z^k\Big) = P(z)Q(z).\end{aligned}$$

Для любой производящей функции $P(z)$ положим

$$\begin{aligned} e(P) &= P'(1), \\ v(P) &= P''(1) + P'(1) = P'(1)^2. \end{aligned}$$

Можно показать, что из соотношений $P(1) = Q(1) = 1$ следуют равенства

$$\begin{aligned} e(PQ) &= e(P) + e(Q), \\ v(PQ) &= v(P) + v(Q). \end{aligned}$$

Доказательство. Имеем

$$\begin{aligned} (PQ)'(z) &= P'(z)Q(z) + P(z)Q'(z), \\ (PQ)''(z) &= P''(z)Q(z) + 2P'(z)Q'(z) + P(z)Q''(z). \end{aligned}$$

Таким образом,

$$\begin{aligned} (PQ)'(1) &= P'(1) + Q'(1) \\ (PQ)''(1) + (PQ)'(1) - (PQ)'(1)^2 &= P''(1) + 2P'(1)Q'(1) + Q''(1) + \\ &\quad + P'(1) + Q'(1) - P'(1)^2 - \\ &\quad - 2P'(1)Q'(1) - Q'(1)^2 = \\ &= P''(1) + P'(1) - P'(1)^2 + \\ &\quad + Q''(1) + Q'(1) - Q'(1)^2. \qquad \square \end{aligned}$$

Производящая функция для сумм $p_k + p_{k+1} + \ldots$

Пусть $q_k = p_k + p_{k+1} + \ldots$ есть вероятность того, что $X \geq k$. Производящая функция $Q(z) = q_0 + q_1 z + q_2 z^2 + \ldots$ не соответствует никакой случайной величине, поскольку $\sum q_k > 1$ за исключением случая, когда $p_0 = 1$.

Поскольку $p_k = q_k - q_{k+1}$, имеем

$$\begin{aligned} P(z) &= \sum_{k\geq 0}(q_k - q_{k+1}) z^k = \\ &= \sum_{k\geq 0} q_k z^k - z^{-1}\sum_{k\geq 0} q_{k+1} z^{k+1} = \\ &= Q(z) - z^{-1}\left(Q(z) - Q(0)\right). \end{aligned}$$

Учитывая, что $Q(0) = 1$, имеем

$$zP(z) = (z-1)Q(z) + 1.$$

Продифференцировав это равенство, получим

$$\begin{aligned} P(z) + zP'(z) &= Q(z) + (z-1)Q'(z), \\ 2P'(z) + zP''(z) &= 2Q'(z) + (z-1)Q''(z), \end{aligned}$$

откуда

$$\begin{aligned} e(P) &= Q(1) - 1, &(1)\\ v(P) &= 2Q'(1) + Q(1) - Q(1)^2. &(2) \end{aligned}$$

2. Дисперсия в задаче собирателя купонов

Обозначим через $P^{(m)}(z)$ производящую функцию следующей случайной величины: “число коробок, которые придётся купить, чтобы получить $(m+1)$-ый купон, когда m купонов из числа n возможных уже собраны”. Обозначим через $P(z)$ производящую функцию случайной величины “число коробок, которые придётся купить, чтобы собрать все n купонов”. Поскольку случайные величины, соответствующие разным значениям m, независимы, имеем

$$P(z) = P^{(0)}(z)P^{(1)}(z)\dots P^{(n-1)}(z),$$

$$\begin{aligned} e(P) &= e(P^{(0)}) + e(P^{(1)}) + \dots + e(P^{(n-1)}), &(3)\\ v(P) &= v(P^{(0)}) + v(P^{(1)}) + \dots + v(P^{(n-1)}). &(4) \end{aligned}$$

Используя формулы на стр. 27, находим

$$\begin{aligned} Q^{(m)}(z) &= 1 + z + \frac{m}{n}z^2 + \left(\frac{m}{n}\right)^2 z^3 + \dots = 1 + \frac{nz}{n-mz}, \\ P^{(m)}(z) &= \frac{(n-m)z}{n-mz}; \\ Q^{(m)}(1) &= 1 + \frac{n}{n-m}, \qquad \left(Q^{(m)}\right)'(1) = \frac{n^2}{(n-m)^2}, \\ e(P^{(m)}) &= \frac{n}{n-m}, \qquad v(P^{(m)}) = \frac{mn}{(n-m)^2}. \end{aligned}$$

Итак, используя формулы (1)–(4), получаем

$$\begin{aligned} e(P) &= \sum_{m=1}^{n} \frac{n}{m} = nH_n = n\ln n + O(n); \\ v(P) &= \sum_{m=1}^{n} \frac{n(n-m)}{m^2} = \\ &= n^2\left(1+\frac{1}{2^2}+\ldots+\frac{1}{n^2}\right) - n\left(1+\frac{1}{2}+\ldots+\frac{1}{n}\right) = \\ &= \frac{\pi^2}{6}n^2 - n\ln n + O(n). \end{aligned}$$

Запись $O(f(n))$ обозначает величину, отношение которой к $f(n)$ остаётся ограниченным при любых достаточно больших n.

Усиление неравенства Чёбышева

Используя в неравенстве Чёбышева тот факт, что среднее квадратичное отклонение $\sqrt{v(P)} = O(n)$, покажем, что число случайных предложений, сделанных мужчинами, страдающими амнезией, с большой вероятностью есть $nH_n + O(n)$. В неравенстве Чёбышева не используется специфика случайной величины, мы же для нашего конкретного случая получим более сильный результат.

Рассмотрим случайную величину с производящей функцией $P^{(m)}(z)$. Её среднее значение равно $n/(n-m)$, среднее квадратичное отклонение — $\sqrt{nm}/(n-m)$. Согласно неравенству Чёбышева при всех $c > 1$ вероятность того, что для получения очередного купона (при наличии m купонов) потребуется купить более, чем $cn/(n-m)$ коробок, не превышает $m/\left(n(c-1)^2\right)$. Точное значение этой вероятности

$$q^{(m)}_{1+cn/(n-m)} = \left(\frac{m}{n}\right)^{cn/(n-m)}.$$

Используя то, что $e^x \geq 1 + x$ при всех x, и полагая в нём $x = (m-n)/n$, получаем оценку

$$q^{(m)}_{1+cn/(n-m)} \leq e^{-c}.$$

При больших значениях c эта оценка лучше, чем неравенство Чёбышева.

Рассмотрим процесс собирания купонов при всех значениях m.

Вероятность того, что потребуется купить более cnH_n коробок, не превышает величины $1 - \prod_{m=0}^{n-1}\left(1 - q^{(m)}_{1+cn/(n-m)}\right)$, откуда, используя приведённое выше неравенство, получаем верхнюю оценку ne^{-c}. Полагая $c = C\ln n$, получаем следующий результат.

Теорема 5. *Вероятность, что для того, чтобы собрать все n купонов, потребуется купить более $CnH_n \ln n$ коробок, не превышает величины $1/n^{C-1}$.*

Оценка для этой вероятности, полученная с помощью неравенства Чёбышева, много грубее, а именно, $1/\left(C^2 \ln^4 n\right)$.

Таким образом, мы показали, что число сделанных случайных предложений редко превосходит nH_n, даже если женихи страдают амнезией. Оценим максимальное количество шагов алгоритма без каких-либо предположений о случайном характере списков предпочтений. (Понятно, что такая задача возникает только при отсутствии амнезии.)

3. Основной алгоритм: исследование наихудшего случая

Разберём решение упражнений из лекции 2.

В худшем случае мужчине придётся “пробежать” по всему списку предпочтений, поскольку выбор им последней женщины в его списке означает остановку алгоритма (все женщины вышли замуж).

Число предложений, сделанных каждым из оставшихся мужчин (их $n-1$), не превышает $(n-1)$. Таким образом, общее число предложений не превышает $n+(n-1)^2 = n^2-n+1$. Ниже приведён пример, когда эта оценка достигается.

A:	b	c	d	c	a	a:	любой список				
B:	c	d	e	b	a	b:	B	C	D	E	A
C:	d	e	b	c	a	c:	C	D	E	A	B
D:	e	b	c	d	a	d:	D	E	A	B	C
E:	b	c	d	e	a	e:	E	A	B	C	D

- A делает предложение b, формируется временная пара Ab.
- B делает предложение c, формируется временная пара Bc.
- C делает предложение d, формируется временная пара Cd.
- D делает предложение e, формируется временная пара De.
- E делает предложение b, формируется временная пара Eb, поскольку b предпочитает E своему партнёру A.
- A снова ищет невесту.
- A делает предложение c, формируется временная пара Ac, поскольку c предпочитает A своему партнёру B.
- B снова ищет невесту.
- B делает предложение d, формируется временная пара Bd, поскольку d предпочитает B своему партнёру C.

- C снова ищет невесту,

и так далее до тех пор пока A не сделает женщине a первое (в её жизни) предложение.

Таким образом, алгоритм остановится, когда A достигнет конца своего списка предпочтений. Мужчины B, C, D, E сделают по 4 предложения каждый и женятся на b, c, d, e, соответственно. Это будет лучшим вариантом для их жён. Каким бы ни был список предпочтений для a, полученное устойчивое паросочетание оптимально для всех женщин.

4. Задача о кратчайшем пути в графе

Пусть имеются города $1, 2, \ldots, n$, между некоторыми из них есть прямое транспортное сообщение. Обозначим длину магистрали, соединяющей города i и j через L_{ij} (будем считать, что $L_{ij} \geq 0$). Положим $L_{ij} = \infty$, если прямое сообщение между городами i и j отсутствует. При этом L_{ij} может не совпадать с L_{ji}.

Э.Дейкстра предложил алгоритм решения важной задачи — поиска кратчайшего пути между городом 1 и любым другим городом. Заметим, что этот алгоритм также применим к решению задач, не связанных с поиском кратчайшего пути, например, с его помощью можно искать минимальное время выполнения всех заданий проекта.

Определим сначала переменные, используемые алгоритмом. Под *расстоянием* от города 1 до любого другого города j будем понимать длину кратчайшего пути между этими городами.

A — множество городов, для которых уже найдено расстояние из города 1.

B — множество всех остальных городов.

d_i — расстояние от города 1 до города i при движении только через города множества A.

k — город множества B, подлежащий перемещению в множество A.

Описание алгоритма

$d_1 \leftarrow 0$; $A \leftarrow \{1\}$; $B \leftarrow \{2, \ldots, n\}$;
for $i \in B$ **do**
    $d_i \leftarrow L_{1i}$;
while B непусто **do**
    begin
    $k \leftarrow$ город в B, такой что $d_k = \min\{d_i : i \in B\}$;
    $B \leftarrow B \setminus \{k\}$; $A \leftarrow A \cup \{k\}$;
    for $i \in B$ **do**
        $d_i \leftarrow \min\{d_i, d_k + L_{ki}\}$
    end;

(В описании алгоритма команда $B \leftarrow B \setminus \{k\}$ вычёркивает элемент k из множества B.)

Пример 1. Исследуем поведение алгоритма на примере, показанном на Рис. 4.

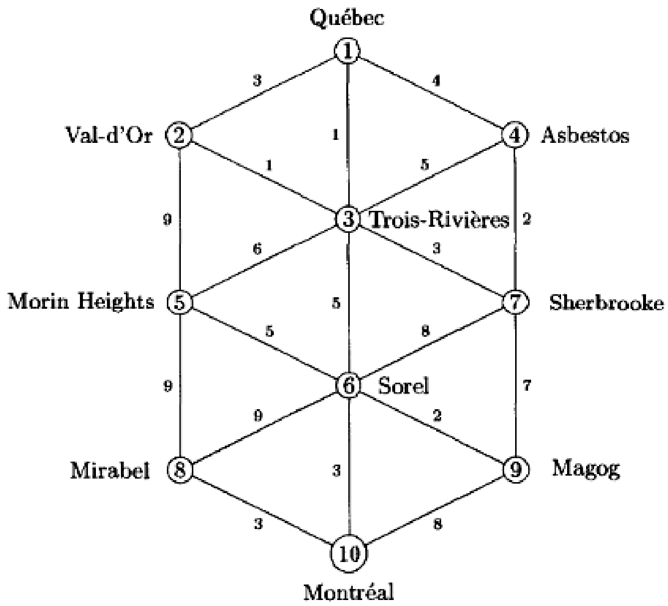


Рис. 4: Задача о кратчайшем пути.

Города пронумерованы от 1 (Квебек) до 10 (Монреаль). Длины магистралей указаны рядом с соответствующими рёбрами графа.

Инициализация. Известен только кратчайший путь из Квебека до него самого:

$$d_1 \leftarrow 0; \quad A \leftarrow \{1\}; B \leftarrow \{2, \ldots, 10\}.$$

Нужно найти кратчайшие пути из Квебека до 9-ти городов, составляющих множество $B = \{2, 3, \ldots, 10\}$.

Только 3 города: Валь-д'Ор, Труа-Ривьер и Асбестос имеют прямое сообщение с Квебеком. Следовательно, $d_2 \leftarrow 3, d_3 \leftarrow 1, d_4 \leftarrow 4$ и $d_i \leftarrow \infty$ для всех остальных городов.

Первый шаг. Поскольку множество B непусто, выполняются команды, заключённые в операторные скобки **begin** ... **end**.

Ближайшим к Квебеку является город Труа-Ривьер. Поэтому

$$k \leftarrow 3, \quad B \leftarrow B \setminus \{3\}, \quad A \leftarrow A \cup \{3\}.$$

(Город с номером 3 вычёркивается из множества B и включается в множество A, поскольку расстояние от Квебека до города номер 3 уже найдено, как будет показано ниже.)

Для всех остальных городов из множества B (то есть $2, 4, 5, \ldots, 10$) выполняется цикл

$$\textbf{for } i \in B \textbf{ do } d_i \leftarrow \min\{d_i, d_3 + L_{3i}\},$$

то есть ищутся кратчайшие из тех маршрутов (из Квебека во все города из множества B), что проходят через города из множества $A = \{1, 3\}$. Получаем таблицу

Города множества B
$d_2 = 2$
$d_4 = 4$
$d_5 = 7$
$d_6 = 6$
$d_7 = 4$

Города множества A
$d_1 = 0$
$d_3 = 1$

(Для всех остальных городов из множества B значения d_i, по прежнему, равны ∞.)

Второй шаг. Множество B всё ещё непусто: $B = \{2, 4, 5, \ldots, 10\}$. Среди всех маршрутов, соединяющих города множества B с Квебеком через города множества A, кратчайшим является маршрут до города 2, поэтому $k \leftarrow 2, \quad B \leftarrow B \setminus \{2\}, \quad A \leftarrow A \cup \{2\}$.

Получаем новую таблицу

Города множества B
$d_4 = 4$
$d_5 = 7$
$d_6 = 6$
$d_7 = 4$

Города множества A
$d_1 = 0$
$d_2 = 2$
$d_3 = 1$

(Маршрут из Морин Хайтс в Квебек, проходящий через Валь-д'Ор, не короче маршрута через Труа-Ривьер, поэтому d_5 не изменяется. Для городов из множества B, не указанных в таблице, значения $d_i = \infty$.)

Третий шаг. Множество B всё ещё непусто: $B = \{4, 5, \ldots, 10\}$. Среди всех маршрутов, соединяющих города множества B с Квебеком через города множества A, кратчайшим является маршрут до города 4, и $k \leftarrow 4, \quad B \leftarrow B \setminus \{4\}, \quad A \leftarrow A \cup \{4\}$.

Мы не приводим полученную таблицу, так как все расстояния $d_i, i \in B$, не изменились. Изменился лишь состав множеств A и B: город 4 переместился из B в A.

Четвертый шаг. $k \leftarrow 7, \quad B \leftarrow B \setminus \{7\}, \quad A \leftarrow A \cup \{7\}$. Теперь мы можем найти кратчайший (в указанном смысле) путь в город 9 ($d_9 = 11$), остальные значения $d_i, i \in B$, не уменьшатся.

Пятый шаг. $k \leftarrow 6, \quad B \leftarrow B \setminus \{6\}, \quad A \leftarrow A \cup \{6\}$. Мы улучшили d_9 и можем достичь городов 8 и 10, что видно из следующей таблицы:

Города множества B	**Города множества A**
$d_5 = 7$	$d_1 = 0$
$d_8 = 15$	$d_2 = 2$
$d_9 = 8$	$d_3 = 1$
$d_{10} = 9$	$d_4 = 4$
	$d_6 = 6$
	$d_7 = 4$

Покажем далее, что в момент перемещения города k из множества B в множество A величина d_k есть *расстояние* из Квебека до города k. Действительно, если бы существовал более короткий путь, то он проходил бы через некоторый город из множества B, отличный от города k. Но все остальные города из множества B не менее удалены от Квебека, чем город k (по определению величины d_k в описании алгоритма), и все дуги графа имеют неотрицательную длину.

Значения $d_i, i = 1, 2, \ldots, 10$, полученные по окончании работы алгоритма, показаны на Рис. 5. Числа, заключённые в рамочки, — это *расстояния*

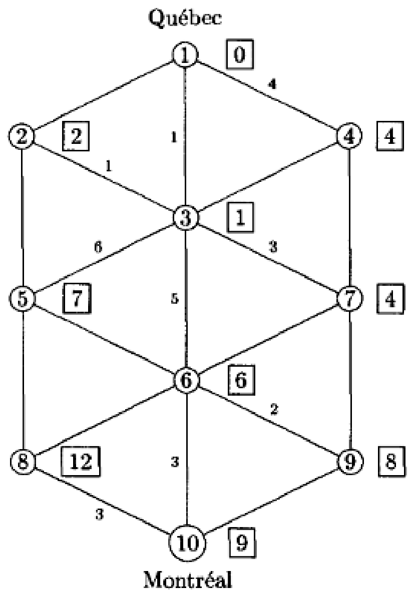


Рис. 5: Расстояния до Квебека.

от Квебека до соответствующего города. Длины указаны только для тех рёбер, которые участвуют в оптимальных маршрутах.

Трудно точно оценить среднее время выполнения алгоритма Дейкстры. Можно предложить следующую модель:

Для каждого города i случайно определим множество городов, имеющих с ним прямое сообщение. Расположив эти города в порядке увеличения L_{ij} (при j, пробегающем соответствующее множество), получим случайную перестановку.

Например, выберем число дуг, исходящих из каждой вершины, по некоторому вероятностному закону. Затем случайным образом выберем пары городов, соединёнными этими дугами. Предположим, что длины этих дуг представляют собой независимые одинаково распределённые случайные величины.

Зададим список предпочтений для каждого города i, располагая остальные города в порядке возрастания величин L_{ij}. Эти списки аналогичны тем, что использовались в задаче об устойчивых паросочетаниях. Таким образом, можно оценить среднее время выполнения алгоритма Дейкстры с помощью результатов, полученных для задачи собирателя купонов.

Можно модифицировать алгоритм Дейкстры, откладывая осуществление операции $d_i \leftarrow \min\{d_i, d_k + L_{ki}\}$ до того момента, когда возникнет необходимость в значении d_i. Получим следующий алгоритм:

$A \leftarrow \{1\}$; $B \leftarrow \{2, \ldots, n\}$;
while B непусто **do**
  begin
  выбрать город i из множества A с непустым списком “предпочтений”;
  $k \leftarrow$ первый город в списке “предпочтений” города i;
  вычеркнуть город k из списка “предпочтений” города i;
  if $k \in B$ **then**
    переместить k из множества B во множество A
  end;

Отметим аналогию с алгоритмом поиска устойчивых паросочетаний: выбор города k аналогичен выбору женщины, которой делается предложение. Важным моментом является то, что значение переменной k выбирается случайно. Тем самым один и тот же подход (использованный в задаче собирателя купонов) применим и к анализу алгоритма поиска устойчивого паросочетания, и для алгоритма Дейкстры. Среднее число шагов модифицированного алгоритма Дейкстры не превышает nH_n. Следовательно, среднее число шагов алгоритма поиска кратчайшего пути от города i до города j для всех i и j (то есть n-кратного применения описанного выше алгоритма) имеет порядок $O(n^2 \ln^2 n)$.

Упражнения

1. Оцените максимальное число предложений, сделанных в алгоритме поиска устойчивого паросочетания, когда все мужчины имеют один и тот же список предпочтений. Списки предпочтений женщин формируются случайным образом. Единственно ли найденное устойчивое паросочетание?

2. Оцените среднее число предложений при следующей матрице предпо-

чтений мужчин:

$$\begin{array}{lllll} A: & a & b & c & d \\ B: & b & c & a & d \\ C: & c & a & b & d \\ D: & a & b & c & d \end{array}$$

и случайной матрице предпочтений женщин.

3. Пусть X — дискретная случайная величина, и пусть p_k — вероятность того, что $X = k$. Производящая функция Дирихле имеет вид

$$P(z) = \sum_{k \geq 1} \frac{p_k}{k^z}.$$

Заметим, что $P(0) = 1$.

(a) Представьте $\mathrm{E}(X)$, $\mathrm{V}(X)$ и $\mathrm{E}(\ln X)$ как функции от $P(z)$ и её производных.

(b) Пусть случайная величина Y не зависит от X и имеет производящую функцию Дирихле $Q(z)$. Найдите производящую функцию Дирихле для произведения $X \cdot Y$.
Представьте $\mathrm{E}(XY)$ и $\mathrm{V}(XY)$ как функции от $\mathrm{E}(X)$, $\mathrm{E}(Y)$, $\mathrm{V}(X)$, $\mathrm{V}(Y)$.

ЛЕКЦИЯ 5

Поиск в хеш-таблицах: поведение основного алгоритма в среднем

Как мы убедились, задача об устойчивых паросочетаниях имеет интересные взаимосвязи с задачей собирателя купонов и задачей о кратчайшем пути. Сейчас мы также увидим её взаимосвязь с фундаментальной задачей поиска информации.

Хеширование

Пусть x — единица информации, чьё расположение в базе данных мы будем искать. Обычно информация в базе данных упорядочена, что обеспечивает возможность её систематического поиска (как, например, информация в телефонном справочнике.)

При работе с компьютерной базой данных эффективна техника, называемая хешированием. Пусть у нас есть n ячеек памяти, где n больше, чем объём данных (то есть некоторые ячейки пусты). Предположим, что множество всевозможных *ключей*[7] очень велико, много больше числа n. (Например, общее число 6-буквенных слов (составленных из букв латинского алфавита) равно 26^6, но только малая часть из них будет использована в программе). Присвоим каждой записи номер из интервала от 1 до n, который будет указывать её положение в базе.

Пусть определена функция $h(x)$, ставящая в соответствие каждой записи x некоторую перестановку на множестве $\{1, 2, \ldots, n\}$, а именно

$$x \to h(x) = (h_1(x), h_2(x), \ldots, h_n(x)).$$

Перестановки для разных x можно считать независимыми. Вероятности получения каждой перестановки одинаковы, то есть равны $1/n!$.

Рассмотрим задачу поиска записи x в таблице. Будем просматривать ячейки с номерами $h_1(x), h_2(x), \ldots$. Мы либо найдем x (успешный поиск), либо попадём в пустую ячейку (безуспешный поиск).

Пример 1. Пусть $n = 9$, и таблица содержит 6 записей:

$$\begin{array}{l}
x_1 \to \mathbf{3}\ 2\ 7\ 8\ 6\ 9\ 1\ 5\ 4 \\
x_2 \to \mathbf{1}\ 5\ 3\ 7\ 6\ 2\ 9\ 4\ 8 \\
x_3 \to \mathbf{4}\ 8\ 1\ 2\ 5\ 9\ 6\ 3\ 7 \\
x_4 \to \mathbf{1}\ \mathbf{5}\ 4\ 9\ 7\ 8\ 3\ 6\ 2 \\
x_5 \to \mathbf{9}\ 8\ 6\ 4\ 1\ 3\ 5\ 7\ 2 \\
x_6 \to \mathbf{4}\ \mathbf{1}\ \mathbf{7}\ 9\ 8\ 6\ 5\ 2\ 3
\end{array}$$

[7] Ключ – часть записи, идентифицирующая её, например, фамилия в телефонном справочнике. Задача заключается в поиске по ключу всей записи целиком; хеш-функция, рассматриваемая ниже, является функцией от ключа — замечание переводчиков.

После вставки этих записей в таблицу имеем

1	2	3	4	5	6	7	8	9
x_2		x_1	x_3	x_4		x_6		x_5

Например, запись x_4 мы найдём в два этапа. Требуемое число шагов равно количеству выделенных жирным шрифтом элементов соответствующей перестановки. Если же, например, мы бы искали запись x_7, которой соответствует перестановка

$$x_7 \to 7\ 1\ 8\ \ldots,$$

то после трёх попыток мы обнаружили бы, что эта запись не внесена в базу (она подлежит размещению в ячейке с номером 8).

Среднее время поиска информации[8]

Пусть в таблицу заносится очередная, $(m+1)$-ая запись. Найдём среднее время, необходимое для поиска свободной ячейки для её размещения. (В дальнейшем это число будет равно среднему числу шагов, необходимому для поиска этой записи.)

Сформулируем этот вопрос иначе: чему равно среднее число безуспешных попыток поиска записи, если общее число записей в таблице равно m?

Пусть $q_k^{(m)}$ — вероятность того, что придётся сделать по крайней мере k шагов. Имеем

$$q_1^{(m)} = 1, \quad q_2^{(m)} = \frac{m}{n}, \quad q_3^{(m)} = \frac{m}{n}\frac{m-1}{n-1}, \quad q_4^{(m)} = \frac{m}{n}\frac{m-1}{n-1}\frac{m-2}{n-2}, \ldots,$$

$$q_k^{(m)} = \frac{m!}{(m-k+1)!}\frac{(n-k+1)!}{n!} = \frac{\binom{n-k+1}{n-m}}{\binom{n}{m}},\ 1 \le k \le m+1,$$

где $\binom{n}{m} = n!/(m!(n-m)!)$ — биномиальный коэффициент. Среднее число шагов, необходимое для размещения $(m+1)$-ой записи равно

$$\begin{aligned} q_1 + q_2 + q_3 + \ldots &= \frac{\binom{n}{n-m} + \binom{n-1}{n-m} + \ldots + \binom{n-m}{n-m}}{\binom{n}{m}} = \\ &= \frac{\binom{n+1}{n-m+1}}{\binom{n}{m}} = \frac{n+1}{n+1-m}. \end{aligned}$$

Если таблица заполнена наполовину, то в среднем двух шагов будет достаточно, чтобы убедиться, что запись отсутствует в таблице.

Дисперсия вычисляется аналогично. Получаем

$$\frac{(n+1)m(n-m)}{(n+1-m)^2(n+2-m)}.$$

[8] Вероятностный анализ в несколько иной модели хеширования рассматривается в разделе 8.5 книги Р. Грэхем, Д. Кнут, О. Паташник, *Конкретная математика*, М.: Мир, 1998 — замечание переводчиков.

К моменту, когда таблица содержит m записей, среднее число шагов успешного поиска записи составляет

$$\frac{1}{m}\left(\frac{n+1}{n+1}+\frac{n+1}{n}+\frac{n+1}{n-1}+\ldots+\frac{n+1}{n+2-m}\right)=\frac{n+1}{m}(H_{n+1}-H_{n+1-m}).$$

Положим $\alpha = m/(n+1)$. Тогда среднее число шагов запишется в виде

$$\frac{1}{\alpha}\ln\left(\frac{1}{1-\alpha}\right)+O\left(\frac{1}{n}\right) \quad \text{при } n\to\infty,$$

поскольку

$$H_{n+1}-H_{n+1-m}=\ln(n+1)-\ln(n+1-m)+O\left(\frac{1}{n}\right).$$

Например, если память занята на 90%, то среднее число шагов составит всего лишь около $10\ln(\frac{10}{9}) \approx 2{,}56$, даже если m и n весьма велики ($m = 900\,000$, $n = 1\,000\,000$).

Связь с проблемой поиска паросочетаний

В лекции 3 мы получили верхнюю оценку среднего числа предложений (числа итераций основного алгоритма). В действительности же это значение может оказаться несколько меньше.

Вычислим среднее число предложений в частном случае, когда все женщины имеют один и тот же список предпочтений, в то время как матрица предпочтений мужчин формируется случайным образом.

Общее число предложений не изменится, если перенумеровать мужчин в другом порядке (поставив на первое место того мужчину, который оказался лучшим для всех женщин, на второе — второго по счёту, и так далее). Тогда каждый кавалер делает женщинам предложения, пока не появится новая женщина. *Алгоритм поиска пары идентичен алгоритму поиска записи в таблице с помощью хеш-функции.*

Вернёмся к примеру 1, предполагая, что число и мужчин, и женщин равно 9. Список предпочтений всех женщин есть $X_1, X_2, \ldots, X_9$. Первые 6 мужчин уже “устроены” — номер занимаемой каждым из них ячейки есть номер его подруги. Мужчина номер 7 делает предложение женщинам 7 и 1, и от обеих получает отказ (ячейки с номерами 7 и 1 уже заняты). Женщина номер 8, получившая первое в жизни предложение, выйдет замуж за X_7.

Таким образом, применимы все результаты предыдущего раздела. Среднее число предложений, сделанных мужчиной с номером $m+1$ составляет $(n+1)/(n+1-m)$. Таким образом, среднее суммарное число предложений равно

$$\begin{aligned}(n+1)\left(\frac{1}{n+1}+\frac{1}{n}+\frac{1}{n-1}+\ldots+\frac{1}{2}\right) &= 1+(n+1)(H_n-1) = \\ &= (n+1)H_n-n.\end{aligned}$$

Это число меньше, чем верхняя оценка, полученная для модели с частичной амнезией, на $n - 2H_n + 1$. Возникает следующее предположение:

Гипотеза. *Среднее число предложений, сделанных в случае, когда предпочтения женщин фиксированы, а предпочтения мужчин случайны, всегда не меньше, чем* $(n+1)H_n - n$.

На самом деле, для случая, когда все женщины имеют один и тот же список предпочтений, судя по всему, справедливо ещё более сильное утверждение. А именно, при всех значениях k вероятность того, что число сделанных предложений окажется не меньше, чем k, достигает в этом случае своего минимального значения. (Это очевидно для $k \leq n+2$ и может быть доказано для $k = n+3$.)

Асимптотическая оценка среднего числа предложений в основном алгоритме

Развивая только что сформулированную гипотезу, можно ожидать, что среднее число предложений будет близко к своей верхней оценке $(n-1)H_n + 1$, а не к гипотетической нижней границе $(n+1)H_n - n$. Доказательство этого факта является наиболее важным результатом этих лекций.

Теорема 6. *При случайных матрицах предпочтений мужчин и женщин среднее число предложений в алгоритме поиска устойчивого паросочетания составляет*

$$nH_n + O\left(\ln^4 n\right).$$

До сих пор мы использовали комбинаторный подход. При доказательстве основной теоремы используется вероятностный метод (называемый также венгерским, поскольку он предложен Эрдёшем, Реньи и др.).[9] Пренебрегая маловероятными случаями, рассмотрим наиболее типичные ситуации.

Пусть $M = 3n\ln^2 n - n$.

Условие C. *Число сделанных предложений не превышает* M. (Причина выбора такого значения M станет ясна позже.)

Обозначим через p вероятность выполнения условия C. Предположим, что все мужчины страдают амнезией. В ходе выполнения алгоритма каждая женщина оценивает мужчину только в тот момент, когда впервые получает предложение от него. В этот момент она случайным образом располагает его среди своих предыдущих кавалеров. Пусть N — число предложений, сделанных мужчинами впервые (не по забывчивости), R — число избыточных (повторно сделанных) предложений. N и R — зависимые

[9] По поводу этого метода см., например, книгу Н. Алон, Дж. Спенсер, *Вероятностный метод*, М.: Бином, 2007 — замечание переводчиков.

случайные величины: с ростом N возрастает и R. Рассматривая задачу собирателя купонов, мы вычислили среднее значение и дисперсию величины $N + R$.

Вероятность того, что $N > M$ (она равна $1 - p$), не превышает вероятности того, что $N + R > M$. Из теоремы 5 (см. стр. 34) получаем $1 - p = O(1/n^2)$, откуда следует оценка для математического ожидания

$$\begin{aligned}
\mathrm{E}(N) &= p \cdot (\mathrm{E}(N) \text{ при условии } C) + (1 - p) \cdot (\mathrm{E}(N) \text{ при ложном } C) \leq \\
&\leq (\mathrm{E}(N) \text{ при выполнении } C) + (1 - p)n^2 = \\
&= (\mathrm{E}(N) \text{ при выполнении } C) + O(1).
\end{aligned}$$

Осталось показать, что среднее число предложений при выполнении условия C равно $nH_n + O(\ln^4 n)$. Это будет следовать из того, что

$$(\mathrm{E}(R) \text{ при выполнении } C) = O(\ln^4 n),$$

поскольку $\mathrm{E}(N) + \mathrm{E}(R) = nH_n$ (см. стр. 27).

Пусть N_A и R_A — количества предложений, сделанных некоторым мужчиной A. Распределение этих предложений не зависит от A. Поэтому $\mathrm{E}(N) = n\,\mathrm{E}(N_A)$ и $\mathrm{E}(R) = n\,\mathrm{E}(R_A)$. Покажем, что среднее число предложений, сделанных мужчиной A, есть $O\big((\ln^4 n)/n\big)$.

Пусть p_k есть вероятность того, что $N_A = k$, при условии C. Если $N_A = k$, то среднее значение R_A равно

$$f(k) = \frac{0}{n} + \frac{1}{n-1} + \ldots + \frac{k-1}{n-(k-1)},$$

поскольку среднее число избыточных предложений, сделанных между m-ым и $(m+1)$-ым "истинными", равно

$$\frac{m}{n} + \left(\frac{m}{n}\right)^2 + \ldots = \frac{m}{n-m}.$$

Среднее число $\mathrm{E}(R_A)$ избыточных предложений при условии C есть

$$\begin{aligned}
\mathrm{E}(R_A) &= \sum_{k \geq 1} p_k f(k) = \\
&= (q_1 - q_2)f(1) + (q_2 - q_3)f(2) + (q_3 - q_4)f(3) + \ldots = \\
&= q_1 f(1) + q_2\big(f(2) - f(1)\big) + q_3\big(f(3) - f(2)\big) + \ldots = \\
&= q_1 \frac{0}{n} + q_2 \frac{1}{n-1} + q_3 \frac{2}{n-2} + \ldots + q_n \frac{n-1}{1},
\end{aligned}$$

где $q_k = p_k + p_{k+1} + \ldots$ и $q_{n+1} = 0$. Чтобы получить верхнюю оценку для $\mathrm{E}(R_A)$, оценим сверху величину q_k (вероятность того, что $N_A \geq k$ при условии C). Пусть ε — нижняя оценка вероятности того, что некоторая (случайно выбранная мужчиной A) женщина a примет его предложение (соответственно, $1 - \varepsilon$ — верхняя оценка вероятности того, что A будет отвергнут женщиной a). Тогда, зная что a уже отвергла $k - 1$ предложение, имеем

$$q_k \leq (1 - \varepsilon)^{k-1}.$$

Обозначим количества предложений, полученных 1-ой, 2-ой, ..., n-ой женщинами от мужчин, отличных от A, через $m_1, m_2, \ldots, m_n$, соответственно. Тогда вероятность того, что женщина, случайно выбранная мужчиной A, примет его предложение, составляет (при условии C)

$$\frac{1}{n}\left(\frac{1}{m_1+1}+\frac{1}{m_2+1}+\ldots+\frac{1}{m_n+1}\right).$$

Функция $1/(m_1+1)+1/(m_2+1)+\ldots+1/(m_n+1)$ достигает своего минимального значения при условиях $m_1+m_2+\ldots+m_n \le M$, $m_i \ge 0$ при всех i, когда $m_i = M/n$, $i = 1, 2, \ldots, n$. Таким образом, вероятность того, что предложение будет принято (в предположении справедливости условия C) составит не меньше

$$\frac{1}{n}\left(\frac{n}{M+n}+\ldots+\frac{n}{M+n}\right)=\frac{n}{M+n}=\frac{1}{3\ln^2 n}.$$

Эту величину примем за значение параметра ε.

Завершение доказательства

Итак, мы хотим доказать верхнюю оценку $\mathrm{E}(R_A) = O\big((\ln^4 n)/n\big)$. Мы знаем, что

$$\mathrm{E}(R_A) \le \sum_{k=1}^{n}(1-\varepsilon)^{k-1}\frac{k-1}{n+1-k},$$

где $\varepsilon = 1/\big(3\ln^2 n\big)$. Используя стандартную технику, завершаем доказательство:

$$\begin{aligned}
\mathrm{E}(R_A) \quad &\le \quad \sum_{1\le k\le n/2}\frac{k(1-\varepsilon)^{k-1}}{n/2}+\sum_{n/2<k\le n} n(1-\varepsilon)^{n/2} \le \\
&\le \quad \frac{2}{n}\sum_{k\ge 1} k\,(1-\varepsilon)^{k-1}+\frac{n^2}{2}\,e^{-\varepsilon n/2} = \\
&= \quad \frac{2}{n\varepsilon^2}+\frac{n^2}{2}\exp\left(\frac{-n}{6\ln^2 n}\right) = O\big((\ln^4 n)/n\big)
\end{aligned}$$

□

Заключение

Анализ среднего числа предложений в алгоритме поиска устойчивых паросочетаний позволил нам проиллюстрировать основные методы анализа алгоритмов:

1. Перебор вариантов и методы работы с конечными суммами.
2. Применение метода производящих функций.
3. Комбинаторный анализ наихудшего случая.
4. Вероятностный метод (называемый также венгерским).

Дополнительное замечание о хешировании

Вычисление случайной перестановки $h_1(x), h_2(x), \ldots, h_n(x)$ слишком трудоёмко на практике. Можно вычислить $h(x)$ и $\delta(x)$, такие что $1 \le h(x) \le n$; $\delta(x)$ и n взаимно просты, и рассмотреть $h(x), h(x)+\delta(x), h(x)+2\delta(x), \ldots, h(x)+(n-1)\delta(x) \pmod n$. Эта процедура называется “двойным хешированием”.

Несмотря на то, что при двойном хешировании используются только арифметические прогрессии, асимптотическое поведение времени поиска информации при нём (как показали Гиба (L. Guibas) и Семереди (E. Szemerédi)) такое же, как и для случайных перестановок. Это утверждение было получено в результате искусного применения вероятностного метода. (Позднее более простое доказательство было получено в работе G. S. Lueker и M. Molodowitch, *More analysis of double hashing*, Combinatorica **13** (1993), 83–96.)

ЛЕКЦИЯ 6

Реализация основного алгоритма

Теперь, когда мы проанализировали основной алгоритм, напишем программу для его компьютерной реализации. Эффективность программы будет обеспечена использованием различных структур данных.

Разбор основного алгоритма, проведённый в лекции 2, показывает, что необходимо предусмотреть выполнение следующих операций:

[1] Присвоение переменной x наилучшего элемента из оставшейся части списка предпочтений мужчины X.

[2] Проверка: предпочитает ли женщина x мужчину X своему жениху?

[3] Помолвка X и x.

[4] Вычёркивание женщины x из списка предпочтений мужчины X.

Операция [1] представляет собой поиск первого элемента x среди оставшихся в списке предпочтений мужчины X, операция [4] — удаление элемента из списка. Выберем структуру, удобную для выполнения этих операций. Представим матрицу предпочтений мужчин в виде двумерной таблицы

$$\begin{pmatrix} HC[1,1] & HC[1,2] & \dots & HC[1,n] \\ \vdots & \vdots & & \vdots \\ HC[n,1] & HC[n,2] & \dots & HC[n,n] \end{pmatrix},$$

где $HC[X,j]$ есть имя женщины, находящейся на j-м месте в списке предпочтений мужчины X.

Мы также будем использовать одномерный массив $H[1],\dots,H[n]$, где $H[X]$ — номер позиции первого из невычеркнутых элементов в списке предпочтений мужчины X. Таким образом, происходит помолвка X с невестой по имени $HC[X,H[X]]$. Операции [1] и [4] выглядят следующим образом:

[1] $x \leftarrow HC[X,H[X]]$;

[4] $H[X] \leftarrow H[X]+1$.

Операция [2] предполагает ответ на вопрос: является ли X более предпочтительным для x, чем её текущий партнёр. Для эффективного решения этой задачи представим списки предпочтений женщин иначе, чем списки предпочтений мужчин.

Пусть число $P[x,X]$ показывает степень предпочтения мужчины X в глазах x: если он — самый желанный для неё, то $P[x,X]=n$, если же он — последний в её списке предпочтений, то $P[x,X]=1$, и $P[x,X]=0$, если $X=\Omega$. Обозначим через $F[x]$ текущего партнёра женщины x. Тогда операция [2] примет вид:

[2] $P[x,X] > P\big[x,F[x]\big]$.

Операция [3] не требует разъяснений. Таким образом, алгоритм выглядит следующим образом:

k, X, x, t: integer;

2: HC: **array**[1:n,1:n] **of** integer;
3: P: **array**[1:n,0:n] **of** integer;
4: H, F: **array**[1:n] **of** integer;
5: Инициализация HC и P;
6: **for** $X \leftarrow 1$ **to** n **do** $H[X] \leftarrow 1$;
7: $k \leftarrow 0$;
8: **for** $x \leftarrow 1$ **to** n **do** $F[x] \leftarrow 0$;
9: **while** $k < n$ **do**
10:   **begin**
11:   $X \leftarrow k + 1$;
12:   **while** $X \neq 0$ **do**
13:     **begin**
14:     $x \leftarrow HC\big[X, H[X]\big]$;
15:     **if** $P[x, X] > P\big[x, F[x]\big]$ **then**
16:       **begin**;
17:       $t \leftarrow F[x]$;
18:       $F[x] \leftarrow X$;
19:       $X \leftarrow t$
20:       **end**;
21:     **if** $X \neq 0$ **then**
22:       $H[X] \leftarrow H[X] + 1$
23:     **end**;
24:   $k \leftarrow k + 1$
25:   **end**;
26: печать решения $\Big(HC\big[1, H[1]\big], \ldots, HC\big[n, H[n]\big]\Big)$

Небольшие модификации

Постараемся улучшить начальный вариант алгоритма за счёт исключения некоторых команд.

(a) Во избежание проверки **if** $X \neq 0$ вставим операцию 22: $H[X] \leftarrow H[X] + 1$ перед командой 14. Из-за этого нам придётся изменить инициализацию H в строке 6: вместо $H[X] \leftarrow 1$ написать $H[X] \leftarrow 0$.

(b) Создадим массив $Q[x] \leftarrow P\big[x, F[X]\big]$, что позволит избежать вычислений значений функций $F[x]$ и $P\big[x, F[X]\big]$ при каждом предложении, сделанном женщине x. Конечно, придётся пересчитывать $Q[x]$ при каждом изменении значения $F[X]$, но это будет происходить нечасто. (Аналогично можно было бы модифицировать операцию $x \leftarrow HC\big[X, H[X]\big]$, но это не даст экономии количества обращений к массивам.)

Инициализация таблицы P

Предположим, что список предпочтений какой-либо женщины имеет вид $(3,2,4,1) = FC[x,\cdot]$. Тогда $P[x,\cdot]$ есть $(1,3,4,2)$. Например, мужчина с номером 3 наиболее предпочтителен для женщины x, поэтому $P[x,3] = 4$. Это можно сделать с помощью двойного цикла

for $x \leftarrow 1$ **to** n **do for** $k \leftarrow 1$ **to** n **do** $P\big[x, FC[x,k]\big] \leftarrow n+1-k$.

Для экономии памяти не будем создавать для P новую матрицу, а модифицируем матрицу FC так, чтобы на ее месте получить P. Очевидно, что каждая строка матрицы предпочтений $FC[x,\cdot]$ есть перестановка. Предлагаемый ниже алгоритм инициализации матрицы P использует циклическую структуру перестановок.[10]

for $x \leftarrow 1$ **to** n **do**
  begin
  for $k \leftarrow 1$ **to** n **do** $FC[x,k] \leftarrow -FC[x,k]$;
  for $k \leftarrow 1$ **to** n **do**
    if $FC[x,X] < 0$ **then**
      begin
      $j \leftarrow k$;
      $x \leftarrow -FC[X,k]$;
      while $X \neq k$ **do**
        begin
        $t \leftarrow -FC[x,X]$;
        $FC[x,X] \leftarrow n+1-j$;
        $j \leftarrow X$;
        $X \leftarrow t$
        end;
      $FC[x,k] \leftarrow n+1-j$
      end
  end

Удобнее в только что приведенной процедуре везде заменить идентификатор FC на P, считая P именем матрицы, в которую осуществляется ввод списков предпочтений женщин. Тогда с учётом модификаций (a) и (b) алгоритм поиска устойчивого паросочетания примет вид:

k, X, x, t: integer;
HC, P: **array**[1:n,1:n] **of** integer;
H, F, Q: **array**[1:n] **of** integer;
Инициализация HC и P;
for $X \leftarrow 1$ **to** n **do** $H[X] \leftarrow 0$;
$k \leftarrow 0$;

---

[10]Перестановка $FC[x,\cdot]$ задает подстановку, обратную к $n+1-P[x,\cdot]$. Любая перестановка представляет собой объединение циклов. Суть алгоритма — мы рассматриваем элементы цикла, в которой входит ранее не рассматриваемый элемент $FC[x,k]$, и меняем ориентацию этого цикла на противоположную — замечание переводчиков.

for $x \leftarrow 1$ **to** n **do** $F[x] \leftarrow Q[x] \leftarrow 0;$
while $k < n$ **do**
  begin
  $X \leftarrow k+1;$
  while $X \neq 0$ **do**
    begin
    $H[X] \leftarrow H[X]+1;$
    $x \leftarrow HC\big[X, H[X]\big];$
    if $P[x,X] > Q[x]$ **then**
      begin
      $t \leftarrow F[x];$
      $F[x] \leftarrow X;$
      $Q[x] \leftarrow P[x,X];$
      $X \leftarrow t$
      end
    end;
  $k \leftarrow k+1$
  end;
печать решения $\big(HC\big[1,H[1]\big], \ldots, HC\big[n,H[n]\big]\big)$

Поиск устойчивого паросочетания, содержащего пару Aa

Модифицируем алгоритм поиска устойчивого паросочетания так, чтобы оно содержало пару Aa и было бы оптимальным для всех мужчин за исключением A. Заметим, что устойчивых паросочетаний с Aa может и не существовать.

Будем использовать результаты первых двух лекций. Применим основной алгоритм для случая, когда число мужчин равно $n-1$ (так как A вычеркнут), модифицируя его так, чтобы учесть следующие две ситуации (см. необходимые и достаточные условия на странице 12):

[1] Не должны рассматриваться пары Bb, если одновременно bAa и AbB (из-за угрозы адюльтера Ab).

[2] Не должны рассматриваться пары Bb, если одновременно BaA и aBb (из-за угрозы адюльтера Ba).

Во избежание ситуации [1] мужчина A согласно модифицированному алгоритму вынужден вычеркнуть из своего списка предпочтений всех женщин, более желанных, чем a; при этом каждая из них в дальнейшем будет отвергать всех, кто менее предпочтителен для неё, чем A:

while $HC\big[A, H[A]\big] \neq a$ **do**
  begin $t \leftarrow HC\big[A, H[A]\big];$
  $Q[t] \leftarrow P[t,A];$
  $H[A] \leftarrow H[A]+1$
  end

Во избежание ситуации [2] алгоритм прекращает работу, когда a принимает предложение от некоего мужчины B такого, что BaA (поскольку в процессе алгоритма a пытается улучшить свой выбор — примечание переводчиков). Модифицированный алгоритм принимает вид:

```
k, X, x, t: integer;
HC, P: array[1:n,1:n] of integer;
H, F, Q: array[1:n] of integer;
Инициализация HC и P;
for X ← 1 to n do H[X] ← 0;
k ← 0;
for x ← 1 to n do F[x] ← Q[x] ← 0;
H[A] ← 1;
while HC[A, H[A]] ≠ a do
  begin
  t ← HC[A, H[A]];
  Q[t] ← P[t, A];
  H[A] ← H[A] + 1
  end;
F[a] ← A;
Q[a] ← P[a, A];
while k < n do
  begin
  X ← k + 1;
  if X ≠ A then
    while X ≠ 0 do
      begin
      H[X] ← H[X] + 1;
      if H[X] > n then
        goto done;
      x ← HC[X, H[x]];
      if P[x, X] > Q[x] then
        begin
        t ← F[x];
        F[x] ← X;
        Q[x] ← P[x, X];
        X ← t;
        if X = A then
          goto done
        end
      end;
  k ← k + 1
  end;
печать решения (HC[1, H[1]], ..., HC[n, H[n]]);
done :
```

Обобщение на случай нескольких вынужденных браков

Будем искать устойчивое паросочетание, в котором мужчины из некоторого подмножества S обязаны жениться на определённых женщинах, причём это решение оптимально для остальных мужчин. Модифицируем алгоритм поиска. Вместо цикла **while** о котором говорилось перед предыдущей программой реализуем следующий алгоритм:

```
for всех A ∈ S do
   begin
   a ← заранее назначенная супруга для A;
   H[A] ← 1;
   while HC[A, H[A]] ≠ a do
      begin
      t ← HC[A, H[A]];
      if Q[t] < P[t, A] then
         begin
         if F[t] ≠ 0 then
            goto done;
         Q[t] ← P[t, A]
         end;
      H[A] ← H[A] + 1
      end;
   if Q[a] > P[a, A] then
      goto done;
   F[a] ← A;
   Q[a] ← P[a, A]
   end;
```

Тогда предыдущая программа (с заменой на только что приведённый код первого цикла **while** вместе с сопутствующими присваиваниями до и после него) решает нашу задачу, если заменить, соответственно, команды “**if** $X \neq A$” и “**if** $X = A$” на “**if** $X \notin S$” и “**if** $X \in S$”.

Поиск справедливого устойчивого паросочетания

Рассмотренные алгоритмы вырабатывают решение, наиболее благоприятное для мужчин (или для женщин после соответствующей модификации обозначений). В наши дни такое неравенство полов недопустимо! Возникает вопрос о решении, в котором представители обоих полов были бы равноправны.

Можно было бы, перечислив все устойчивые паросочетания, сделать выбор из них в соответствии с некоторым критерием “справедливости”. Однако при большом числе решений такой подход может оказаться слишком трудоёмким. Вообще говоря, количество устойчивых паросочетаний заранее неизвестно. Поэтому представляется более предпочтительным использование другого подхода.

Например, выберем пару Aa случайным образом. Выполним модифицированный алгоритм поиска устойчивого паросочетания, содержащего пару Aa. При существовании такого решения выполним те же действия для задачи размерности $n-1$ (исключив из исходных данных A и a). В противном случае назначаем невесте a другого жениха и повторяем весь процесс.

Можно улучшить алгоритм так, чтобы число обращений к процедуре поиска устойчивого паросочетания не превышало бы n^2. Тогда среднее число шагов алгоритма будет иметь полиномиальный порядок $O(n^4)$ с модой $O(n^3 \ln n)$.

С. Селков (Stan Selkow) предложил алгоритм, где случайный выбор играет меньшую роль. Он вырабатывает *справедливое оптимальное решение* в том смысле, что оно минимизирует неудовлетворённость самого несчастного. В паросочетании M определим меру неудовлетворённости конкретного человека как номер позиции партнера в рейтинге рассматриваемого человека. Обозначим максимальную среди $2n$ значений меры неудовлетворённости через $U(M)$. Наша цель — минимизировать $U(M)$. На первом шаге алгоритма находятся (с помощью основного алгоритма) как решение, оптимальное для всех мужчин, так и решение, оптимальное для всех женщин. Они дают верхнюю и нижнюю оценки меры неудовлетворённости $U(M)$ (в качестве иллюстрации полезны таблицы на стр. 64 для данных со стр. 60 — примечание переводчиков). Составляем пары из индивидов, чьи верхние и нижние оценки совпадают. Повторяем процесс для задачи меньшей размерности, пока это возможно. Затем (в редуцированной системе) среди индивидов с максимальными верхними оценками случайным образом выбираем одного. Без ограничения общности, считаем, что это — женщина (иначе поменяем местами роли мужчин и женщин). Обозначим её через a. Пусть A — худший вариант для a из всех оставшихся мужчин. Заметим, что (в силу оптимальности решения по Парето) a представляет собой наилучшую кандидатуру для A (среди всех оставшихся женщин). Исключаем a из списка предпочтений A и получаем новое решение оптимальное для мужчин. При этом увеличивается значение нижней оценки меры неудовлетворённости мужчины A и ещё, по крайней мере, у одного мужчины (у которого он “отобьёт” свою нынешнюю невесту — примечание переводчиков). Одновременно снижается верхняя оценка меры неудовлетворённости женщины a и, по крайней мере, еще одной женщины. Очевидно, что повторяя этот процесс, мы получим устойчивое паросочетание.

Поиск всех устойчивых паросочетаний

В этом разделе мы опишем *рекурсивную процедуру* “*toutm*”, зависящую от параметра j, пробегающего значения от 0 до n. В начале работы процедуры $(H[1], H[2], \ldots, H[n])$ — некоторое устойчивое паросочетание. По завершении работы алгоритма эти переменные $H[k]$ примут своё первоначальное значение. Результатом будет печать всех устойчивых паросочетаний $(H'[1], H'[2], \ldots, H'[n])$ таких что

$$H'[k] \begin{cases} = H[k] & \text{для } 1 \le k \le j, \\ \ge H[k] & \text{для } j < k \le n. \end{cases}$$

(Никто из мужчин не улучшает свой выбор, причём мужчины с номерами от 1 до j не меняют своих партнёрш.)

Для получения всех устойчивых паросочетаний мы сначала выполняем основной алгоритм, но в момент вывода решения, оптимального для мужчин, запускаем рекурсивную процедуру: $toutm(0)$.

Процедура $toutm$ имеет вид:

```
procedure toutm(j: integer);
x, X, y, t: integer;
SH, SF, SQ: array[1:n] of integer;
if j = n then печать паросочетания (H[1], ..., H[n])
else begin for t ← 1 to n do
     begin SH[t] ← H[t];
     SF[t] ← F[t];
     SQ[t] ← Q[t]
     end
continue:   toutm(j + 1);
change:     X ← j + 1;
            y ← HC[X, H[X]]; (запрещаем X жениться на этой даме)
propose:    H[X] ← H[X] + 1;
            if H[X] > n then goto finish;
            x ← HC[X, H[X]];
            if P[x, X] > Q[x] then
              begin t ← F[x];
              if t ≤ j then goto finish;
              F[x] ← X;
              Q[x] ← P[x, X];
              if x = y then goto continue;
              X ← t
              end;
            goto propose;
finish:     for t ← 1 to n do
              begin H[t] ← SH[t];
              F[t] ← SF[t];
              Q[t] ← SQ[t]
              end
   end.
```

Корректность этого алгоритма доказать труднее, чем корректность основного алгоритма, хотя здесь применим то же подход. Будем считать, что по предположению индукции вызов процедуры $toutm$(j+1) корректен.

Сначала покажем, что к моменту выполнения операции $H[X] \leftarrow H[X]+1$ все устойчивые паросочетания, где партнёршей X является $H[X]$, уже получены. (Тем самым ни одно устойчивое паросочетание не будет пропущено.)

Затем убедимся в том, что все полученные паросочетания устойчивы. Действительно, если некоторый мужчина A предпочитает женщину a своей текущей партнёрше, то и для неё кто-то предпочтительнее A, поскольку в процессе выполнения алгоритма женщины улучшают свой выбор.

Нерекурсивная версия алгоритма поиска всех устойчивых паросочетаний

Нетрудно преобразовать приведённый выше алгоритм к нерекурсивному виду:

```
begin
x, X, y, t, j: integer;
SH, SF, SQ: array[0:n, 1:n] of integer;
HC, P: array[1:n, 1:n] of integer;
H, F, Q: array[1:n] of integer;
j ← 0;
toutm: if j = n then печать паросочетания (H[1], ..., H[n]);
else
  begin for t ← 1 to n do
    begin SH[j,t] ← H[t];
    SF[j,t] ← F[t];
    SQ[j,t] ← Q[t]
    end;
  continue: j ← j + 1;
  goto toutm;
  change: (выполнение предыдущего набора команд)
  propose: (выполнение предыдущего набора команд)
  finish: for t ← 1 to n do
    begin H[t] ← SH[j,t];
    F[t] ← SF[j,t];
    Q[t] ← SQ[j,t]
    end
  end;
j ← j − 1;
if j ≥ 0 then goto change
end
```

Можно показать, что время выполнения этой программы есть $O(n^3S)$, где S — число устойчивых паросочетаний. (Улучшенная версия алгоритма, требующая лишь $O(n^2 + nS)$ шагов была позднее представлена в публикации Дэна Гасфилда — Dan Gusfield, *Three fast algorithms for four problems in stable marriage*, SIAM J. Comput. **16** (1987), 111–128.)

ЛЕКЦИЯ 7

Нерешённые проблемы

Задачи, поставленные ниже, хотя и далеки от уровня знаменитых проблем Гильберта, сформулированных в его знаменитой лекции в Париже в 1900 году, всё же заслуживают внимания и, вероятно, являются разрешимыми.

Проблема 1. Исследовать среднее число смены партнёров женщиной в процессе выполнения основного алгоритма (женщина x меняет жениха всякий раз, когда меняется $F[x]$).

Будет ли это число существенно меньше числа предложений? Кажется естественным предположение о том, что это число имеет порядок $n \ln \ln n$. В самом деле, женщина со случайным списком предпочтений всегда принимает первое предложение, она принимает второе предложение с вероятностью $1/2, \ldots$, k-е предложение — с вероятностью $1/k$. Женщина, получившая k предложений, меняет женихов в среднем $1+1/2+1/3+\ldots+1/k = H_k$ раз. В среднем женщина получает H_n предложений, поэтому среднее количество смен партнёра ею должно быть примерно равно $\ln \ln n$.

Проблема 2. В лекции 5 мы выдвинули гипотезу о нижней оценке для $\mathrm{E}(N)$ — среднего числа принятых предложений при известной матрице предпочтений женщин. Верна ли эта гипотеза?

Проблема 3. Пусть при фиксированной матрице предпочтений мужчин предпочтения женщин формируются случайно (см. лекцию 5). Верна ли гипотеза о том, что среднее число предложений достигает своего максимума, когда все мужчины имеют одинаковые списки предпочтений?

Проблема 4. Существует ли эффективный способ точного вычисления среднего числа предложений при фиксированной матрице предпочтений женщин? Иначе говоря, можно ли точно определить это значение по структуре матрице предпочтений женщин? Возможно, оно зависит от некоторых свойств матрицы предпочтений таких, например, как число несовпадений рейтингов мужчин в списках предпочтений женщин. Тем самым мы получили бы элегантное решение упражнения 1 к лекции 3.

Проблема 5. Найти матрицы предпочтений n мужчин и n женщин, при которых число устойчивых паросочетаний максимально.

Например, при $n = 4$ максимальное из известных значений числа устойчивых паросочетаний равно 10. Эта ситуация возникает при следующих матрицах предпочтений:

$$
\begin{array}{lcccc}
A: & a & b & c & d \\
B: & b & a & d & c \\
C: & c & d & a & b \\
D: & d & c & b & a
\end{array}
\qquad
\begin{array}{lcccc}
a: & D & C & B & A \\
b: & C & D & A & B \\
c: & B & A & D & C \\
d: & A & B & C & D
\end{array}
\quad .
$$

Здесь в устойчивых паросочетаниях партнёршами A, B, C и D являются, соответственно, дамы

$$\begin{array}{cccc} a & b & c & d \\ b & a & c & d \\ a & b & d & c \\ b & a & d & c \\ b & d & a & c \\ c & a & d & b \\ c & d & a & b \\ c & d & b & a \\ d & c & a & b \\ d & c & b & a \end{array} .$$

Проблема 6. Найти способ описания структуры множества устойчивых паросочетаний при заданных матрицах предпочтений, который позволил бы охарактеризовать решения, не прибегая к их явному нахождению.

Замечания к проблеме 6. Согласно следующим результатам, полученным Джоном Конвеем (John Conway), такие структуры существуют, хотя многие вопросы остаются ещё нерешёнными.

Теорема 7. *Пусть $M = (Aa, Bb, \ldots, Zz)$ и $M' = (Aa', Bb', \ldots, Zz')$ — два произвольных устойчивых паросочетания. Тогда паросочетание*

$$M \vee M' = \Big(A \max_A(a, a'), B \max_B(b, b'), \ldots, Z \max_Z(z, z'),\Big)$$

также устойчиво.
(Здесь $\max\limits_A(a, a')$ есть женщина, наиболее предпочтительная для A из двух кандидатур a и a'.)

Доказательство. Покажем вначале, что $M \vee M'$ есть паросочетание, то есть никакая женщина не имеет одновременно двух мужей.

Предположим, что

$$a = \max_A(a, a') = \max_B(b, b') = b'.$$

Тогда в паросочетании M' мужчина B женат на a. Так как паросочетание M' устойчиво и aAa', имеем BaA. Но паросочетание M также устойчиво. Тогда bBa, что противоречит $\max\limits_B(b, a) = a$.

Покажем теперь, что паросочетание $M \vee M'$ устойчиво. Пусть, например, мужчина A и женщина $\max\limits_B(b, b')$ предпочитают друг друга своим супругам в паросочетании $M \vee M'$. Тогда для A кандидатура b предпочтительнее, чем $\max\limits_A(a, a')$, то есть bAa. Учитывая, что AbB, паросочетание M не является устойчивым. Противоречие.

Следствие 1. *В условиях предыдущей теоремы паросочетание $M \wedge M' = \Big(A \min\limits_A(a, a'), B \min\limits_B(b, b'), \ldots, Z \min\limits_Z(z, z'),\Big)$ также устойчиво.*

Доказательство. Достаточно поменять мужчин и женщин местами. По теореме 3 из лекции 2 паросочетание, о котором будет идти речь в модифицированной таким образом теореме 7, совпадёт с паросочетанием из следствия 1. □

Операции $\vee$ ("max") и $\wedge$ ("min") являются ассоциативными, коммутативными, идемпотентными и дистрибутивными:

$$(M \wedge M') \wedge M'' = M \wedge (M' \wedge M''),$$
$$(M \wedge M') \vee M'' = (M \vee M')' \wedge (M' \vee M''),$$
$$M \wedge M = M,$$

и т.д.

Таким образом, справедливо

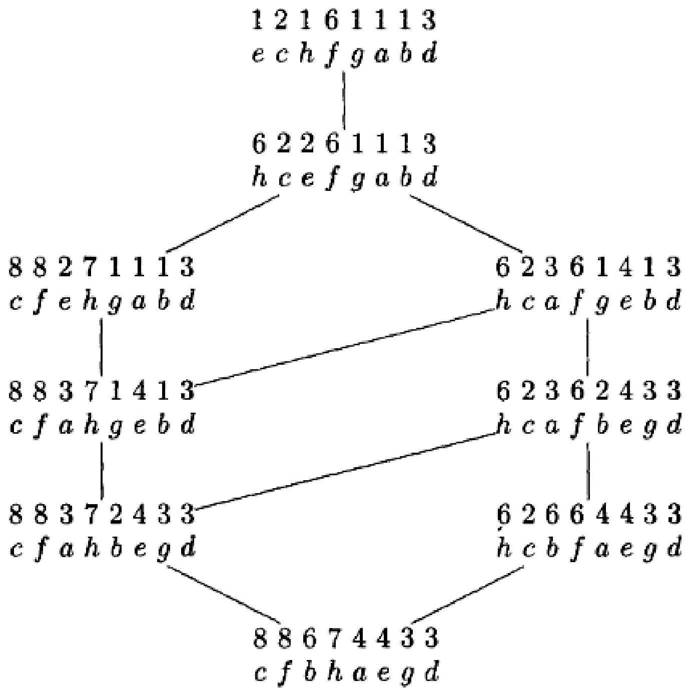
Следствие 2. *При любых матрицах предпочтений мужчин и женщин множество устойчивых паросочетаний представляет собой дистрибутивную решётку.*

Для мужчин оптимальным решением является "максимум" по всем устойчивым паросочетаниям, для женщин – "минимум".

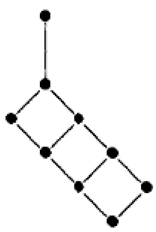
Рассмотрим решётку устойчивых паросочетаний для нескольких случаев. Известен следующий пример, где число мужчин и число женщин равны 8:

$$\begin{array}{lcccccccc}
A: & e & g & a & b & f & h & d & c \\
B: & b & c & g & e & d & a & h & f \\
C: & h & e & a & d & f & b & c & g \\
D: & c & b & g & d & a & f & h & e \\
E: & g & b & e & a & c & f & h & d \\
F: & a & f & g & e & h & d & b & c \\
G: & b & e & g & f & c & d & h & a \\
H: & c & h & d & e & g & b & f & a
\end{array}
\qquad
\begin{array}{lcccccccc}
a: & E & C & G & F & A & B & H & D \\
b: & H & F & C & E & G & B & A & D \\
c: & A & E & F & B & D & H & G & C \\
d: & H & G & C & B & D & A & E & F \\
e: & F & D & G & C & H & A & B & E \\
f: & B & H & E & D & F & C & G & A \\
g: & G & E & B & A & H & F & D & C \\
h: & G & D & A & E & B & C & F & H
\end{array}.$$

В соответствии с приведённой ниже решёткой можно ввести отношение порядка на множестве из 9-ти устойчивых паросочетаний. (Паросочетания представлены как списки партнёрш мужчин A, B, C, D, E, F, G, H, соответственно; число над каждым именем есть позиция каждой женщины в списке предпочтений её супруга.)

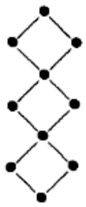


Эта структура представима как следующая дистрибутивная решётка:

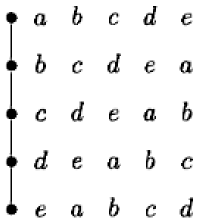


Очевидно, что при движении по решётке сверху вниз позиции жён в списке предпочтений их мужей не убывают.

Ниже приведена решётка для 10 устойчивых паросочетаний, рассмотренных в проблеме 5.

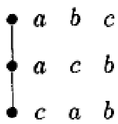


Решётка для n устойчивых паросочетаний, рассмотренных в примере 3 лекции 1 имеет вид:



Еще один пример решётки в виде цепи, но во всех соседних паросочетаниях есть одинаковые пары.

$$\begin{array}{llll} A: & a & c & b \\ B: & b & c & a \\ C: & c & b & a \end{array} \qquad \begin{array}{llll} a: & B & A & C \\ b: & C & B & A \\ c: & A & B & C \end{array}$$



Далее пример решётки с более сложной структурой:

$$\begin{array}{lllllll} A: & a & b & c & d & e & f \\ B: & b & c & a & e & f & d \\ C: & c & a & b & f & d & e \\ D: & d & e & f & a & b & c \\ E: & e & f & d & b & c & a \\ F: & f & d & e & c & a & b \end{array} \qquad \begin{array}{lllllll} a: & E & F & D & B & C & A \\ b: & F & D & E & C & A & B \\ c: & D & E & F & A & B & C \\ d: & B & C & A & E & F & D \\ e: & C & A & B & F & D & E \\ f: & A & B & C & D & E & F \end{array} \quad .$$

Можно убедиться в том, что

$$\begin{array}{lllllll} Aa & \Rightarrow & Cc & \Rightarrow & Bb & \Rightarrow & Aa, \\ Ab & \Rightarrow & Ca & \Rightarrow & Bc & \Rightarrow & Ab, \\ Af & \Rightarrow & Ce & \Rightarrow & Bd & \Rightarrow & Af, \\ Ae & \Rightarrow & Cd & \Rightarrow & Bf & \Rightarrow & Ae, \end{array} \qquad \begin{array}{lllllll} Dd & \Rightarrow & Ff & \Rightarrow & Ee & \Rightarrow & Dd, \\ De & \Rightarrow & Fd & \Rightarrow & Ef & \Rightarrow & De, \\ Dc & \Rightarrow & Fb & \Rightarrow & Ea & \Rightarrow & Dc, \\ Db & \Rightarrow & Fa & \Rightarrow & Ec & \Rightarrow & Db. \end{array}$$

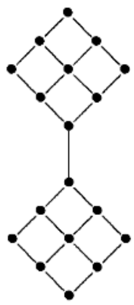
Далее,

$$\begin{array}{rcl@{\qquad}rcl} Ac & \Rightarrow & Ff, Fd \text{ или } Fe; & Ac, Ff & \Rightarrow & Ee, Dd, Ba, Cb; \\ Ac, Fd & \Rightarrow & Ef, De, Ba, Cb; & Ac, Fe & \Rightarrow & Ba, Df, Cb, Ed. \end{array}$$

Таким образом,

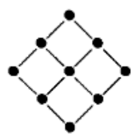
$$Ac \Rightarrow Ba \Rightarrow Cb \Rightarrow Ac,$$

и решётка устойчивых паросочетаний имеет следующую структуру:



Эту решётку можно представить как *прямую сумму* решётки

и двух решёток вида



Последнее частично упорядоченное множество является *прямым произведением* двух решёток вида

Возникает вопрос: можно ли построить все дистрибутивные решётки с помощью всевозможных матриц предпочтений? Какими общими свойствами должны обладать решётки, порождённые латинскими квадратами, такими как в упражнении 3 лекции 1? (Такой тип имеют многие из приведённых выше примеров.)

Решётки могут также использоваться в другом обобщении задачи об устойчивом паросочетании: можно ослабить отношение порядка, заменив списки предпочтений мужчин и женщин решётками. Но это может привести к потере структуры частично упорядоченного множества на множестве всех устойчивых паросочетаний. Действительно, пусть все мужчины и все женщины ранжируют лиц противоположного пола одинаково: мужчины — согласно решётке

$$\begin{tikzcd}[column sep=small] & a \arrow[dl, dash] \arrow[dr, dash] & \\ b \arrow[dr, dash] & & c \arrow[dl, dash] \\ & d & \end{tikzcd}$$

а женщины — согласно решётке

$$\begin{tikzcd}[column sep=small] & A \arrow[dl, dash] \arrow[dr, dash] & \\ B \arrow[dr, dash] & & C \arrow[dl, dash] \\ & D & \end{tikzcd}$$

Тогда результат операции “$\vee$” для двух устойчивых паросочетаний $M = (Aa, Bb, Cc, Dd)$ и $M' = (Aa, Bc, Cb, Dd)$, то есть $M \vee M' = (Aa, Ba, Ca, Dd)$, не будет паросочетанием.

Пересечение интервалов

Применяя основной алгоритм для поиска решения, оптимального для мужчин, и для поиска решения, оптимального для женщин, мы получим в списке предпочтений каждого мужчины и каждой женщины границы “интервалов” выбора. В любой паре любого устойчивого паросочетания каждый из супругов находится в соответствующем интервале своего партнёра.

Рассмотрим приведённый выше пример для 8-ми мужчин и 8-ми женщин.

Здесь ♂ обозначает решение оптимальное для мужчин, ♀ — решение оптимальное для женщин, “—” означает, что женщина находится за пределами интервала выбора мужчины, и, наконец, “/” означает, что мужчины находится за пределами интервала выбора женщины. (Последние два символа не используются в матрице предпочтений женщин, поскольку их отсутствие не скажется на работе обсуждаемых ниже алгоритмов — примечание переводчиков.)

Например, мужчина A не может жениться на b, так как он не попадает в её интервал выбора.

$$
\begin{array}{lcccccccc}
A: & e^{\text{♂}} & \cancel{g} & \cancel{a} & \cancel{b} & \cancel{f} & h & \cancel{d} & c_{\text{♀}} \\
B: & \sout{b} & c^{\text{♂}} & \cancel{g} & \cancel{e} & \cancel{d} & \cancel{a} & h & f_{\text{♀}} \\
C: & h^{\text{♂}} & e & a & \cancel{d} & \cancel{f} & b_{\text{♀}} & \sout{e} & \sout{g} \\
D: & \sout{e} & \sout{b} & \sout{g} & \sout{d} & \sout{a} & f^{\text{♂}} & h_{\text{♀}} & \sout{c} \\
E: & g^{\text{♂}} & b & \cancel{c} & a_{\text{♀}} & \sout{e} & \sout{f} & \sout{h} & \sout{d} \\
F: & a^{\text{♂}} & \cancel{f} & \cancel{g} & e_{\text{♀}} & \sout{h} & \sout{d} & \sout{b} & \sout{c} \\
G: & b^{\text{♂}} & e & g_{\text{♀}} & \sout{f} & \sout{c} & \sout{d} & \sout{h} & \sout{a} \\
H: & \sout{c} & \sout{h} & d^{\text{♂}}_{\text{♀}} & \sout{e} & \sout{g} & \sout{b} & \sout{f} & \sout{a}
\end{array}
$$

$$
\begin{array}{lcccccccc}
a: & E_{\text{♀}} & C & G & F^{\text{♂}} & A & B & H & D \\
b: & H & F & C_{\text{♀}} & E & G^{\text{♂}} & B & A & D \\
c: & A_{\text{♀}} & E & F & B^{\text{♂}} & D & H & G & C \\
d: & H^{\text{♂}}_{\text{♀}} & G & C & B & D & A & E & F \\
e: & F_{\text{♀}} & D & G & C & H & A^{\text{♂}} & B & E \\
f: & B_{\text{♀}} & H & E & D^{\text{♂}} & F & C & G & A \\
g: & G_{\text{♀}} & E^{\text{♂}} & B & A & H & F & D & C \\
h: & G & D_{\text{♀}} & A & E & B & C^{\text{♂}} & F & H
\end{array}
$$

Это ограничение существенно повышает скорость алгоритмов поиска всех устойчивых паросочетаний и поиска “справедливых” паросочетаний, описанных в лекции 6. Однако в некоторых исключительных случаях сузить перебор не удаётся — когда оптимальным для мужчин является решение, в котором их супруги занимают первые места в их списках предпочтений, в то время как сами мужья являются наименее желанными партнёрами для своих жён, и наоборот.

Пересечение интервалов является необходимым, но не достаточным условием устойчивости. В приведённом выше примере пересечение интервалов содержит паросочетание Ac, Bh, Ce, Df, Eb, Fa, Gg, Hd, однако пара Bh не входит ни в одно устойчивое паросочетание.

Проблема 7. При заданных случайным образом матрицах предпочтений найти асимптотическое поведение среднего числа устойчивых паросочетаний.[11] При $n = 2$ это число равно $1\frac{1}{8}$, при $n = 3$ оно равно $1\frac{1139}{3888}$.[12] В общем случае это число равно $n!p_n$, где p_n есть вероятность того, что паросочетание $A_1a_1, A_2a_2, \dots, A_na_n$ является устойчивым. Имеем

$$p_n = \frac{1}{n!^n} \sum_{\{x_{ij}\}} \prod_{j=1}^{n} \frac{r_j!(n-1-r_j)!}{1+c_j} \tag{5}$$

$$p_n = \sum_{\{x_{ij}\}} \prod_{j=1}^{n} \frac{(-1)^{r_j}}{(1+r_j)(1+c_j)}, \tag{6}$$

где в обоих случаях суммирование происходит по всем $2^{n(n-1)}$ матрицам из элементов x_{ij}, равных 0 или 1, причём диагональные элементы равны 0. Здесь $r_i = \sum_j x_{ij}$ — сумма элементов i-ой строки, а $c_j = \sum_i x_{ij}$ — сумма элементов j-го столбца.

Формула (5) получена непосредственным перечислением матриц предпочтений, таких что паросочетание $A_1a_1, A_2a_2, \dots, A_na_n$ устойчиво, $a_jA_ia_i \Leftrightarrow x_{ij} = 1$. Здесь $r_j!(n-1-r_j)!$ — количество j-х строк матриц предпочтений мужчин, соответствующих заданной матрице (x_{ij}), при этом $(1+c_j)^{-1}$ — доля матриц предпочтений женщин, для которых указанное паросочетание устойчиво.

Для получения соотношения (6) воспользуемся формулой включения-исключения, рассматривая события $E_{ij} = \{$имеет место адюльтер $A_ia_j\}$. Каждое из $2^{n(n-1)}$ слагаемых формулы включения-исключения есть вероятность совместного наступления некоторых событий $E_{ij}, E_{lk}, \dots$. Сопоставим этому слагаемому матрицу X, где на соответствующих местах $((i,j),(l,k),\dots$ находятся единицы, а остальные позиции заняты нулями. Соответствующая вероятность находится из соображений, что пара A_ia_i является *одновременно* худшей (по сравнению с теми, с которыми имеют место адюльтеры) как для мужчины A_i, так и для женщины a_i. Вероятности соответствующих событий задаются формулами $(1+r_i)^{-1}$ (для мужчины A_i) и $(1+c_i)^{-1}$ (для женщины a_i). Знак каждого слагаемого в сумме (6) определяется чётностью количества единиц в матрице X.[13] (Интересно доказать эквивалентность формул (5) и (6) алгебраическим путём.)

Ещё одна формула для p_n имеет вид:

$$p_n = \int_0^1 \int_0^1 \dots \int_0^1 dx_1\, dx_2 \dots dx_n\, dy_2 \dots dy_n \prod_{\substack{i,j=1\\ i\neq j}}^{n} (1 - x_iy_j). \tag{7}$$

[11]Главный член асимптотики был найден Б. Питтелем (Pittel, B.) в статье *The Average Number of Stable Matchings*, SIAM J. Discrete Math. **2**(4), 530-549 (1989) (подробнее см. в диссертации C. Lennon *On the likely number of stable marriages*, The Ohio State University, 2007) — замечание переводчиков.

[12]Компьютерные расчёты по формуле (6) показывают, что при $n = 4$ получается $1\frac{323027}{663552}$, при $n = 5$ среднее число устойчивых паросочетаний есть $1\frac{32683287021803}{46656000000000}$, а при $n = 6$ оно равно $1\frac{34755368731928869}{37324800000000000}$ — замечание переводчиков.

[13]Более подробное доказательство формулы (6) имеется в упомянутой выше диссертации Крейга Леннона – замечание переводчиков.

Например,

$$p_3 = \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 da\,db\,dc\,dA\,dB\,dC\,(1-Ab)\times$$
$$\times(1-Ac)(1-Ba)(1-Bc)(1-Ca)(1-Cb).$$

Доказательство. Раскрыв произведение в формуле (7)

$$\prod_{\substack{i,j=1\\ i\neq j}}^{n}(1-x_iy_j) = \sum_{\{x_{ij}\}}\prod_{j=1}^{n}(-x_j)^{r_j}y_j^{c_j},$$

мы получаем формулу (6). □

Решение проблемы 7 с помощью этих формул неочевидно. Мы знаем, что величина $n!p_n$ должна быть не меньше 1, поскольку, по крайней мере, одно устойчивое паросочетание всегда существует, однако, это не следует непосредственно из формул (5)–(7).

Проблема 8. Пусть даны матрицы предпочтений мужчин и женщин. Построить ориентированный граф "разводов" (см. лекцию 1). В нём будет $n!$ вершин, каждая из которых соответствует перестановке $(\pi_1,\ldots,\pi_n)$ элементов множества $\{1,2,\ldots,n\}$ (номеров женщин в паросочетании с фиксированным порядком мужчин).

Пусть $(A_1a_{\pi_1},\ldots,A_na_{\pi_n})$ — неустойчивое паросочетание. Предположим, что при некоторых $i\neq j$ имеют место предпочтения $a_{\pi_j}A_ia_{\pi_i}$ и $A_ia_{\pi_j}A_j$ (то есть имеет место адюльтер $A_ia_{\pi_j}$). Тогда граф разводов содержит дугу из вершины $(\pi_1,\ldots,\pi_n)$ в вершину $(\pi_1',\ldots,\pi_n')$, где $\pi_i'=\pi_j$, $\pi_j'=\pi_i$, а для всех остальных номеров k имеет место совпадение $\pi_k'=\pi_k$. Вершины, не имеющие исходящих дуг, соответствуют устойчивым паросочетаниям.

Всегда ли существует путь, начинающийся в заданной вершине и приводящий к устойчивому паросочетанию? Если да, то насколько он может быть короток?[14]

Возможно, этот ориентированный граф имеет интересные свойства, связанные с частично упорядоченным множеством из проблемы 6.

Проблема 9. Существует ли алгоритм поиска устойчивого паросочетания, где количество операций, даже в наихудшем случае, растёт медленнее, чем n^2? (Мы не учитываем времени ввода матриц предпочтений мужчин и женщин.)[15]

Проблема 10. Существует ли интересная взаимосвязь между задачей об устойчивых паросочетаниях и задачей о назначении? (Последняя состоит в том, чтобы при заданной матрице из вещественных элементов a_{ij} найти перестановку π, максимизирующую сумму $\sum_{i=1}^{n} a_{i\pi_i}$.)

[14] См. далее замечание к работе [2] на стр. 72 — замечание переводчиков.

[15] Эта проблема решена (см. Cheng Ng, Daniel S. Hirschberg, *Lower Bounds for the Stable Marriage Problem and its Variants*, SIAM J. Comput. **19**(1): 71–77 (1990)) — замечание переводчиков.

Проблема 11. Допускает ли задача об устойчивых паросочетаниях обобщение на случай трёх множеств (например, мужчины, женщины, собаки)?[16]

Проблема 12. (Однополый вариант задачи: устойчивость расселения по двуместным комнатам в общежитии.) Каждый индивид в группе, состоящей из $2n$ человек, составляет рейтинг оставшихся $2n-1$ потенциальных соседей. Найти эффективный (полиномиальный в наихудшем случае) алгоритм поиска устойчивого сочетания (разбиения $2n$ человек на n пар).[17]

Например, можно убедиться в том, что при матрице предпочтений

$$\begin{array}{lccccccc}
A: & B & E & D & F & G & H & C \\
B: & C & F & A & G & H & E & D \\
C: & D & G & B & H & E & F & A \\
D: & A & H & C & E & F & G & B \\
E: & F & A & H & B & C & D & G \\
F: & G & B & E & C & D & A & H \\
G: & H & C & F & D & A & B & E \\
H: & E & D & G & A & B & C & F
\end{array}$$

имеется лишь 3 устойчивых решения: $\{AB, CD, FG, HE\}$, $\{BC, DA, EF, GH\}$, $\{AE, BF, CG, DH\}$.

Иногда устойчивых решений не существует, например,

$$\begin{array}{llll}
A: & B & C & D \\
B: & C & A & D \\
C: & A & B & D \\
D: & \multicolumn{3}{l}{\text{что угодно}}
\end{array}$$

(Любой, кто окажется соседом D, сможет подыскать себе лучший вариант.) Возможно, что проблема распознавания существования устойчивого решения для этой задачи NP-полна.

[16] См. по этому поводу работы: [1] E. Boros, V. Gurvich, S. Jaslar, D. Krasner, *Stable matchings in three-sided systems with cyclic preferences*, Discrete Math. **289** (2004), 1–10; [2] Péter Biró and Eric McDermid, *Three-Sided Stable Matchings with Cyclic Preferences*, Algorithmica **58** (2010), 5–18 и ссылки там — замечание переводчиков.

[17] Эта проблема решена Р. Ирвингом (Irving, R., *An efficient algorithm for the "stable room-mates" problem*, J. Algorithms **6**, 577–595. (1985)) — замечание переводчиков.

Резюме лекций

Задача об устойчивых паросочетаниях позволила нам познакомиться с разнообразными методами анализа алгоритмов. Мы использовали понятия из различных областей математики:

- Комбинаторики,
- Теории вероятности,
- Математического анализа,
- Алгебры,

и информатики:

- Структуры данных,
- Управляющие структуры,
- Вычислительная сложность.

Многие замечательные проблемы всё ещё остаются нерешенными.

Аннотированная библиография

Устойчивые паросочетания

1. T.H.F. Brissenden, *Some derivations from the marriage bureau problem*, Math. Gaz. **58**, 250–257 (1974). (Есть ошибки.)

2. Juan Bulnesm, Jacobo Valdes, *Notes on the complexity of the stable marriage problem*, Stanford University, 1972 (неопубликовано). (Материалы использованы в упражнениях 1 и 2 лекции 2.)

3. D. Gale, L.S. Shapley, *College admissions and the stability of marriage*, Amer. Math. Monthly **69**, 9–15 (1962). (Классическая оригинальная статья.)

4. D.G. McVitie, L.B. Wilson, *Stable marriage assignment for unequal sets*, BIT **10**, 295–309 (1970).

5. D.G. McVitie, L.B. Wilson, *The application of the stable marriage to university admissions*, Oper. Res. Quart. **21**, 425–433 (1970).

6. D.G. McVitie, L.B. Wilson, *The stable marriage problem*, Comm. ACM **14**, 486–490 (1971). (Нахождение всех устойчивых паросочетаний.)

7. L.B. Wilson, *An analysis of the stable marriage assignment algorithm*, BIT **12**, 569–575 (1972). (Верхняя оценка для случая частичной амнезии.)

Задача собирателя купонов

1. R.E. Greenwood, *Coupon collector's test for random digits*, Math. Tables Aids Comput. **9**, 1–5 (1955).

2. D.E. Knuth, *Seminumerical Algorithms. The Art of Computer Programming*, v. 2, Addison-Wesley, Reading, Mass., 1969, Ch. 3.3.2 E, Exercises 7–10. (Имеется русский перевод: Д.Э. Кнут, Искусство программирования, Том.2, Получисленные алгоритмы, М.: Вильямс, 2007, глава 3.3.2 E, упражнения 7–10, см. также Д.Э. Кнут, Искусство программирования для ЭВМ, Том.2, Получисленные алгоритмы, М.: Мир, 1978.)

Задача о кратчайшем пути

1. E.W. Dijkstra, *A note on two problems in connection with graphs*, Numer. Math. **1**, 269–271 (1959).

2. P.M. Spira, *A new algorithm for finding all shortest paths in a graph of positive arcs in average time* $O(n^2 \log^2 n)$, SIAM J. Comput. **2**, 28–32 (1973).

Хеширование

1. L.I. Guibas, *The analysis of hashing algorithms*, Ph.D. Thesis, Stanford University, Computer Science Dept., 1976.
2. D.E. Knuth, *Sorting and Searching. The Art of Computer Programming*, vol. 3, Addison-Wesley, Reading, Mass., 1973, Section 6.4 (Имеется русский перевод: Д.Э. Кнут, Искусство программирования, Том 3, Сортировка и поиск, М.: Вильямс, 2008, глава 6.4, см. также Д.Э. Кнут, Искусство программирования для ЭВМ, Том. 3, Сортировка и поиск, М.: Мир, 1977.)
3. W.W. Peterson, *Addressing for random-access storage*, IBM J. Res. Develop. **1**, 130–146 (1957).
4. J.D. Ullman, *A note on the efficiency of hashing functions*, J. Assoc. Comput. Mach. **19**, 569–575 (1972).

Структуры данных и алгоритмы

1. D.E. Knuth, *Fundamental Algorithms. The Art of Computer Programming*, vol. 1, Addison-Wesley, Reading, Mass., 1968 (Имеется русский перевод: Д.Э. Кнут, Искусство программирования, Том 1, Структуры данных и алгоритмы, М.: Вильямс, 2008, см. также Д.Э. Кнут, Искусство программирования для ЭВМ, Том.1, Структуры данных и алгоритмы, М.: Мир, 1976.)
2. D.E. Knuth, *Structured programming with **go to** statements*, Comput. Surveys **6**, 261–301 (1974).

Анализ алгоритмов

1. D.E. Knuth, *The analysis of algorithms*, Actes du Congres International des Mathematiciens (Nice, 1970), T. 3, Gauthier-Villars, Paris, 1971, pp. 269–274.
2. D.E. Knuth, *Mathematical analysis of algorithms*, Information Processing **71** (Proc. IFIP Congress 1971, Ljubljana, 1971), vol. 1, North-Holland, Amsterdam, 1972, pp. 19–27.
3. R. Sedgewick, *Quicksort*, Ph.D. Thesis, Stanford University, Computer Science Dept., 1975.

ПРИЛОЖЕНИЕ А
Более поздние работы

1. Dan Gusfield, Robert W. Irving, *The Stable Marriage Problem: Structure and Algorithms*, MIT Press Series in the Foundations of Computing, MIT Press, Cambridge, Mass., 1989, 240 pp.
   (Со времени публикации этих лекций теория устойчивых паросочетаний получила существенное развитие. В упомянутой книге содержится полный обзор результатов, полученных к 1989 году, включая состояние дел в отношении 12 проблем, сформулированных в лекции 7.)

2. Akihisa Tamura, *Transformation from arbitrary matchings to stable matchings*, J. Combin. Theory Ser. A **62**, (1993), 310–323.
   (Отрицательный ответ на вопрос проблемы 8: устойчивость не всегда достигается с помощью последовательности разводов.)

3. Fred Galvin, *The list chromatic index of a bipartite multigraph*, J. Combin. Theory Ser. B **63**, (1995), 153–158.[18]
   (Теория устойчивых паросочетаний применяется к знаменитой гипотезе Диница (J. Dinitz).)

4. Donald E. Knuth, Rajeev Motwani, Boris Pittel, *Stable husbands*, Random Structures Algorithms **1**, (1990), 1–14.
   (Показано, что при случайном формировании списков предпочтений женщин число супругов каждой женщины находится в пределах от $(\ln n)/2$ до $\ln n$ с вероятностью, стремящейся к 1 при $n \to \infty$. Упрощена техника вероятностного анализа. Введено понятие “динамическое связывание” как синоним понятия “принцип отложенных решений”.)

5. Donald E. Knuth, *An exact analysis of stable allocation*, Journal of Algorithms **20** (2), (1996), 431–442.
   (Интересные результаты о перестановках получаются при рассмотрении задачи о размещении заказов неделимой продукции среди её поставщиков. Статья содержит сведения о текущем положении дел в решении проблемы 2, остающейся и по сей день одной из самых привлекательных проблем в анализе основного алгоритма поиска устойчивых паросочетаний.)

6. Tomás Feder, *Stable Network and Product Graphs*, Mem. Amer. Math. Soc., vol. 555, (1995), 223 pp.
   (Полная монография, где теория устойчивых паросочетаний рассматривается в более общем контексте устойчивых конфигураций в сети.

[18] Имеется изложение на русском языке в книге М. Айгнер, Г. Циглер, *Доказательства из книги*, М.: Мир, 2006, стр. 197—202 — замечание переводчиков.

Получено много элегантных результатов, в том числе новые алгоритмы, вырабатывающие симметричные (относительно мужчин и женщин) решения. Например, устойчивое паросочетание с минимальной "степенью неудовлетворённости" (где ранг самого нежеланного партнёра принимает минимально возможное значение) можно найти за $O(n^2)$ шагов. "Справедливое" устойчивое паросочетание (то есть такое, где минимальна сумма рангов всех партнёров) можно найти за $O(n^3)$ шагов.)

ПРИЛОЖЕНИЕ В

Ответы и решения

Упражнения к лекции 1

1. (a) Докажем это утверждение от противного. Пусть устойчивое паросочетание $(A_j a_j, A_k b, \ldots)$ оптимально для A_j, а паросочетание $(A_j c, A_k a_j, \ldots)$ оптимально для A_k. Это означает, что $a_j A_k b$, откуда $A_j a_j A_k$. С другой стороны, $a_j A_j c$, откуда $A_k a_j A_j$. Противоречие.

   (b) Предположим противное. Пусть $a_k A_j a_j$ и $A_j a_k A_k$, то есть паросочетание, в котором есть пары $A_k a_k$ и $A_j a_j$, неустойчиво из-за возможности измены со стороны A_j и a_k. Предположим, что устойчивое оптимальное для A_k паросочетание имеет вид $(A_k a_k, A_j b, \ldots)$. Тогда по определению a_j имеет место предпочтение $a_j A_j b$ (так как по предположению $b \neq a_j$). Учитывая свойство транзитивности предпочтений, имеем $a_k A_j b$ и, учитывая $A_j a_k A_k$, получаем противоречие с устойчивостью паросочетания $(A_k a_k, A_j b, \ldots)$.

2. (Ae, Ba, Cb, Dc, Ed) и (Aa, Bb, Cc, Dd, Ee).

Упражнения к лекции 2

1. и 2. См. лекцию 4, часть 3.

Упражнения к лекции 3

2. Без ограничения общности предположим, что AdC. Число напрасно сделанных предложений принимает значения $0, 1, 2, 3$ с вероятностями $1/9, 7/18, 1/4, 1/4$, соответственно. Например, напрасных предложений не будет, если BbC, DbB и CcD (вероятность этого равна $1/12$) или же если CbB, DbC, BcD и CcB (вероятность этого равна $1/36$).

Упражнения к лекции 4

1. Пусть $a, b, c, \ldots$ — общий список предпочтений всех мужчин. Женщина a сделает лучшую для себя партию, например, A. Остальные мужчины сделают предложение даме b. Её выбор, скажем, B, также будет наилучшим (если BbA) либо вторым в её рейтинге. Остальные мужчины сделают предложение даме c, чей выбор, также будет наилучшим из множества женихов без A и B, и так далее. Общее число предложений не превышает $n + (n-1) + \ldots + 1 = n(n+1)/2$. Решение единственно, так как оно оптимально для всех мужчин и всех женщин. Заметим, что среднее число предложений было бы меньше nH_n, если инициатива принадлежала бы женщинам (см. теорему 4 — замечание переводчиков).

2. Порядок предложений (независимо от матрицы предпочтений женщин) таков: $a, b, c, a, b, c, \ldots$, пока не будет сделано предложение даме d. Вероятности того, что будет сделано 7, 8, 9, 10, 11, 12, 13 предложений, равны, соответственно, $1/4, 1/6, 2/9, 25/108, 1/12, 1/36, 1/54$. Среднее значение равно $8\frac{8}{9}$. (См. также проблему 3 в лекции 7.)

3. (a) $\mathrm{E}(X) = P(-1)$, $\mathrm{V}(X) = P(-2) - P(-1)^2$, $\mathrm{E}(\ln X) = -P'(0)$.

   (b) Производящая функция есть $P(z)Q(z)$, $\mathrm{E}(XY) = \mathrm{E}(X)\mathrm{E}(Y)$, $\mathrm{V}(XY) = \bigl(\mathrm{V}(X) + \mathrm{E}(X)^2\bigr)\bigl(\mathrm{V}(Y) + \mathrm{E}(Y)^2\bigr) - \mathrm{E}(X)^2\mathrm{E}(Y)^2$.

Указатель