

Оглавление

Предисловие	12
О чем эти книги.....	12
Навыки, которые вы приобретете	14
В чем особенность этой книги.....	15
Для кого эта книга?	15
Дополнительные ресурсы	16
Благодарности	17
От издательства	18
Глава 19. Что такое NP-трудность?	19
19.1. Задача о минимальном остовном дереве и задача коммивояжера: алгоритмическая загадка.....	20
19.1.1. Задача о минимальном остовном дереве.....	20
19.1.2. Задача коммивояжера.....	21
19.1.3. Безуспешные попытки решить задачу коммивояжера.....	23
19.1.4. Решения к тестовым заданиям 19.1–19.2.....	26
19.2. Возможные уровни профессиональной компетенции	27
19.3. «Легкие» и «трудные» задачи.....	28
19.3.1. Полиномиально-временные алгоритмы	29
19.3.2. Полиномиальное время против экспоненциального.....	30
19.3.3. Легкорешаемые задачи	31
19.3.4. Относительная труднорешаемость	33
19.3.5. Трудные задачи	33
19.3.6. Предположение, что $P \neq NP$	35
19.3.7. Предварительное определение NP-трудности.....	36

19.3.8. Рандомизированные и квантовые алгоритмы	36
19.3.9. Тонкости	37
19.4. Алгоритмические стратегии для NP-трудных задач.....	38
19.4.1. Универсальный, правильный, быстрый (выбрать два).....	38
19.4.2. Компромисс в отношении универсальности.....	40
19.4.3. Компромисс в отношении правильности.....	41
19.4.4. Компромисс в отношении скорости	43
19.4.5. Ключевые выводы.....	44
19.5. Доказательство NP-трудности: простой рецепт	44
19.5.1. Редукции.....	45
19.5.2. Использование упрощений для разработки быстрых алгоритмов.....	46
19.5.3. Использование упрощений для распространения NP-трудности	48
19.5.4. NP-трудность бесцикловых кратчайших путей.....	49
19.5.5. Решение к тестовому заданию 19.3	54
19.6. Ошибки новичков и допустимые неточности	55
Задачи на закрепление материала.....	59
Задачи повышенной сложности	61
Задача по программированию	62
Глава 20. Компромисс в отношении правильности: эффективные неточные алгоритмы	63
20.1. Минимизация производственной продолжительности	63
20.1.1. Определение задачи	64
20.1.2. Жадные алгоритмы	66
20.1.3. Алгоритм Грэма.....	66
20.1.4. Время работы.....	68
20.1.5. Приближенная правильность	69
20.1.6. Доказательство теоремы 20.1	71
20.1.7. Сначала самое длительное время обработки.....	73
20.1.8. Доказательство теоремы 20.4	76
20.1.9. Решения к тестовым заданиям 20.1–20.3.....	77
20.2. Максимальный охват	80
20.2.1. Определение задачи	80
20.2.2. Дальнейшие применения	82

20.2.3. Жадный алгоритм	82
20.2.4. Плохие примеры для GreedyCoverage	84
20.2.5. Приближенная правильность	87
20.2.6. Ключевая лемма	87
20.2.7. Доказательство теоремы 20.7.....	90
20.2.8. Решения к тестовым заданиям 20.4–20.5.....	91
*20.3. Максимизация влияния.....	92
20.3.1. Каскады в социальных сетях	92
20.3.2. Пример	93
20.3.3. Задача о максимизации влияния	94
20.3.4. Жадный алгоритм	96
20.3.5. Приближенная правильность	97
20.3.6. Влияние есть взвешенная сумма функций охвата.....	97
20.3.7. Доказательство теоремы 20.9.....	98
20.3.8. Решение к тестовому заданию 20.6.....	100
20.4. Эвристический алгоритм двукратной замены для задачи коммивояжера ...	101
20.4.1. Решение задачи коммивояжера.....	101
20.4.2. Улучшение тура двукратной заменой	103
20.4.3. Алгоритм двукратной замены 2-OPT.....	105
20.4.4. Время работы.....	107
20.4.5. Качество решения	108
20.4.6. Решения к тестовым заданиям 20.7–20.8.....	108
20.5. Принципы локального поиска.....	109
20.5.1. Метаграф допустимых решений	110
20.5.2. Парадигма проектирования алгоритма локального поиска	111
20.5.3. Три модельных решения	113
20.5.4. Два решения по проектированию алгоритма	116
20.5.5. Время выполнения и качество решения	117
20.5.6. Избегание плохих локальных оптимумов.....	118
20.5.7. Когда использовать локальный поиск?	119
20.5.8. Решения к тестовым заданиям 20.9–20.10.....	120
Задачи на закрепление материала.....	122
Задачи повышенной сложности	126
Задачи по программированию	132

Глава 21. Компромисс в отношении скорости:**точные неэффективные алгоритмы 134**

21.1. Алгоритм Беллмана — Хелда — Карпа для задачи коммивояжера	134
21.1.1. Базовый уровень: исчерпывающий поиск.....	134
21.1.2. Динамическое программирование.....	136
21.1.3. Оптимальная подструктура	137
21.1.4. Рекуррентное уравнение	139
21.1.5. Подзадачи.....	140
21.1.6. Алгоритм Беллмана — Хелда — Карпа	141
21.1.7. Решение к тестовому заданию 21.1	143
*21.2. Поиск длинных путей посредством цветового кодирования.....	144
21.2.1. Актуальность.....	144
21.2.2. Определение задачи	145
21.2.3. Первая атака на подзадачи.....	146
21.2.4. Цветовое кодирование	148
21.2.5. Вычисление самого дешевого панхроматического пути	149
21.2.6. Правильность и время выполнения	152
21.2.7. Рандомизация спешит на помощь.....	153
21.2.8. Окончательный алгоритм	156
21.2.9. Время выполнения и правильность	157
21.2.10. Пересмотр сетей белок-белковых взаимодействий	158
21.2.11. Решения к тестовым заданиям 21.2–21.4.....	159
21.3. Алгоритмы для конкретных задач против волшебных ящиков	160
21.3.1. Редукции и волшебные ящики.....	160
21.3.2. Решатели задач MIP и SAT	161
21.3.3. Чему вы научитесь и чему не научитесь.....	162
21.3.4. Ошибки новичка повторно	162
21.4. Решатели задач MIP	163
21.4.1. Пример: задача о рюкзаке	163
21.4.2. Задачи MIP в более общем случае.....	165
21.4.3. Решатели задач MIP: некоторые отправные точки	167
21.5. Решатели задач SAT	168
21.5.1. Пример: раскраска графа	168

21.5.2. Выполнимость булевых формул	170
21.5.3. Кодирование раскраски графа как задачи SAT	171
21.5.4. Решатели задач SAT: некоторые отправные точки.....	173
Задачи на закрепление материала.....	175
Задачи повышенной сложности	178
Задачи по программированию	182
Глава 22. Доказательство NP-трудных задач.....	184
22.1. Редукции повторно.....	184
22.2. Задача 3-SAT и теорема Кука — Левина	187
22.3. Общая картина.....	189
22.3.1. Ошибка новичка повторно	189
22.3.2. Восемнадцать редукций	190
22.3.3. Зачем продираться через доказательства NP-трудности?	192
22.3.4. Решение к тестовому заданию 22.1	193
22.4. Шаблон для редукций	193
22.5. Задача о независимом множестве является NP-трудной	195
22.5.1. Главная идея	196
22.5.2. Доказательство теоремы 22.2	198
*22.6. Ориентированный гамильтонов путь является NP-трудным	201
22.6.1. Кодирование переменных и присвоение истинности	201
22.6.2. Кодирование ограничений	203
22.6.3. Доказательство теоремы 22.4	205
22.7. Задача коммивояжера является NP-трудной	207
22.7.1. Задача о неориентированном гамильтоновом пути.....	207
22.7.2. Доказательство теоремы 22.7.....	208
22.8. Задача о сумме подмножества является NP-трудной	211
22.8.1. Базовый подход	212
22.8.2. Пример: четырехвершинный цикл.....	213
22.8.3. Пример: пятивершинный цикл	213
22.8.4. Доказательство теоремы 22.9	214
Задачи на закрепление материала.....	217
Задачи повышенной сложности	220

Глава 23. P, NP и все такое прочее.....	222
*23.1. Накопление свидетельств труднорешаемости.....	223
23.1.1. Построение убедительного случая с помощью редукций.....	223
23.1.2. Выборка множества C для задачи коммивояжера.....	224
*23.2. Решение, поиск и оптимизация	226
*23.3. NP: задачи с легко распознаваемыми решениями.....	227
23.3.1. Определение класса сложности NP	227
23.3.2. Примеры задач в NP.....	228
23.3.3. Задачи NP поддаются решению исчерпывающим поиском	229
23.3.4. NP-трудные задачи.....	230
23.3.5. Теорема Кука — Левина повторно.....	231
23.3.6. Решение к тестовому заданию 23.1	234
*23.4. Предположение, что $P \neq NP$	234
23.4.1. P: задачи NP, поддающиеся решению за полиномиальное время ...	234
23.4.2. Формальное определение предположения	235
23.4.3. Статус предположения, что $P \neq NP$	236
*23.5. Гипотеза об экспоненциальном времени	238
23.5.1. Требуют ли NP-трудные задачи экспоненциального времени?.....	238
23.5.2. Сильная гипотеза об экспоненциальном времени.....	239
23.5.3. Нижние границы времени выполнения для простых задач.....	240
*23.6. NP-полнота.....	243
23.6.1. Редукции Левина	243
23.6.2. Самые трудные задачи в NP	245
23.6.3. Существование NP-полных задач	246
Задачи на закрепление материала.....	248
Задачи повышенной сложности	249
Глава 24. Практический пример: стимулирующий аукцион FCC.....	251
24.1. Перенацеливание беспроводного спектра	252
24.1.1. От телевидения к мобильным телефонам.....	252
24.1.2. Недавнее перераспределение спектра	253
24.2. Жадные эвристики для выкупа лицензий	254
24.2.1. Четыре временных упрощающих допущения.....	255
24.2.2. Засада со стороны взвешенного независимого множества.....	256

24.2.3. Жадные эвристические алгоритмы	257
24.2.4. Многоканальный случай.....	260
24.2.5. Засада со стороны раскраски графа	261
24.2.6. Решение к тестовому заданию 24.1	262
24.3. Проверка допустимости.....	263
24.3.1. Кодирование в качестве задачи выполнимости	263
24.3.2. Встраивание реберных ограничений	264
24.3.3. Задача о переупаковке.....	265
24.3.4. Трюк № 1: предварительные решатели (в поисках легкого выхода)	266
24.3.5. Трюк № 2: предварительная обработка и упрощение	268
24.3.6. Трюк № 3: портфель решателей задач SAT.....	270
24.3.7. Терпимость к отказам.....	271
24.3.8. Решение к тестовому заданию 24.2	271
24.4. Реализация в виде нисходящего тактового аукциона	272
24.4.1. Аукционы и алгоритмы.....	272
24.4.2. Пример	273
24.4.3. Переосмысление жадного алгоритма FCCGreedy	275
24.4.4. Пора получить компенсацию	277
24.5. Окончательный результат	278
Задачи на закрепление материала.....	280
Задача повышенной сложности	282
Задача по программированию	282
Эпилог: полевое руководство по разработке алгоритмов	283
Подсказки и решения	286
Книги Тима Рафгардена.....	299

*20.3. Максимизация влияния

GreedyCoverage (раздел 20.2) изначально задумывался для классических применений, таких как выбор местоположений для новых заводов. В XIX веке обобщения этого алгоритма нашли новое применение в computer science. Этот раздел описывает репрезентативный пример в анализе социальных сетей.¹

20.3.1. Каскады в социальных сетях

Для наших целей *социальная сеть* — это ориентированный граф $G = (V, E)$, в котором вершины соответствуют пользователям, а ориентированное ребро (v, w) означает, что v «влияет на» w . Например, w следит за новостями v .

ПРОСТАЯ КАСКАДНАЯ МОДЕЛЬ

Изначально каждая затравочная вершина является «активной», а все остальные вершины «неактивными». Все ребра изначально находятся «не в подброшенном» состоянии.

Пока есть активная вершина v и не подброшенное исходящее ребро (v, w) , надо:

- подбросить несмещенную монету: орел выпадает с вероятностью p ;
 - если выпадает орел, обновить статус ребра (v, w) значением «активное». Если вершина w не активна, обновить статус ребра значением «активное»;
 - если выпадает решка, обновить статус ребра (v, w) значением «неактивное».
-

Каскадная модель показывает, как информация (например, новостная статья или мем) перемещается по социальной сети. Вот простой пример сети,

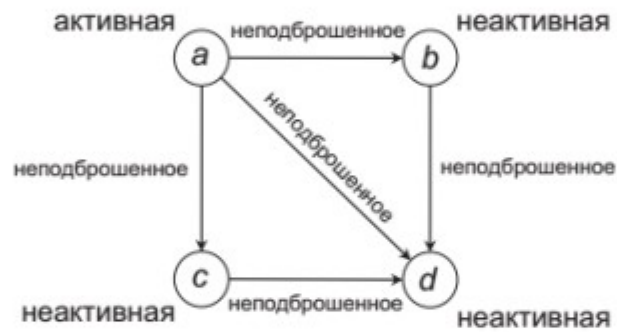
¹ Разделы, помеченные звездочкой, подобные этому, являются более трудными, и при первом прочтении их можно пропустить.

параметризованной ориентированным графом $G = (V, E)$, активационной вероятностью $p \in [0, 1]$ и подмножеством $S \subseteq V$ *затравочных* (seed) вершин.¹

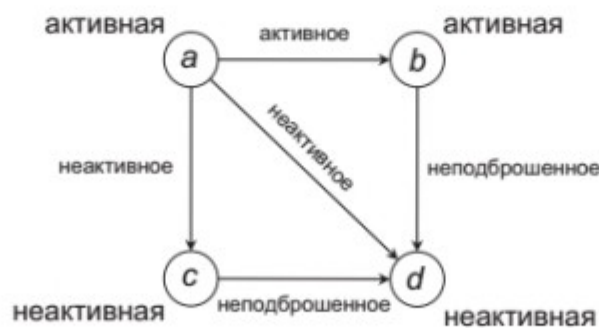
После того как вершина активирована (скажем, вследствие чтения статьи или просмотра фильма), она больше не станет неактивной. Вершина может иметь несколько активационных возможностей — одну для каждого ее активизированного влиятельного объекта. Например, первые две рекомендации от друзей относительно нового фильма не регистрируются, но третья побуждает вас его посмотреть.

20.3.2. Пример

В графе

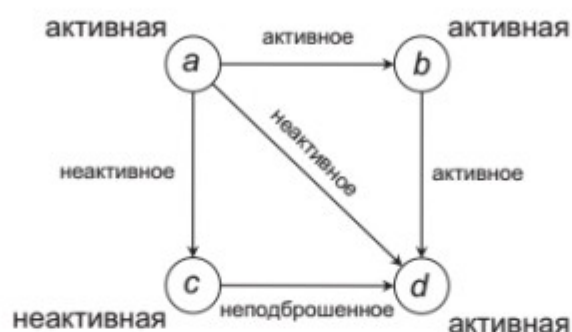


вершина a является затравочной и изначально активной. Остальные вершины изначально неактивны. Каждое из исходящих ребер (a, b) , (a, c) и (a, d) имеет вероятность p активирования другой конечной точки ребра. Предположим, что монета, связанная с ребром (a, b) , выпадает орлом, а две другие — решкой. Новая картина такова:

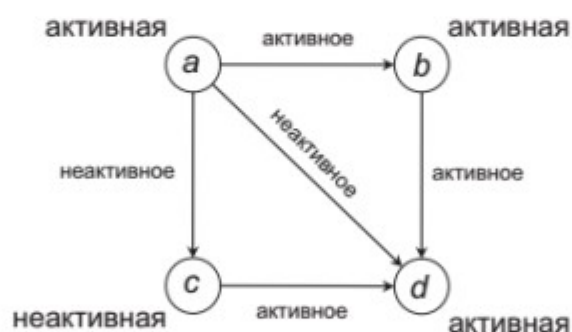


¹ Для того чтобы освежить информацию о базовой дискретной вероятности, см. Приложение Б *Первой части* либо ресурсы на веб-сайте www.algorithmsilluminated.org.

В этот момент нет никакой надежды активировать вершину c . Остается вероятность p , что вершина d активируется через неподброшенное ребро (b, d) . Если это событие произойдет, то финальным состоянием будет:



Для замыкания и удобства можно факультативно добавить шаг постобработки, который подбрасывает монеты любым оставшимся неподброшенным ребрам и соответствующим образом обновляет их статусы (оставляя статусы всех вершин неизменными). В нашем рабочем примере окончательный результат может быть таким:



В общем случае, с шагом постобработки или без него, вершины, которые в итоге активируются в конце процесса, достигаются из затравочной вершины по ориентированному пути, состоящему из активированных ребер.

20.3.3. Задача о максимизации влияния

Цель задачи о максимизации влияния состоит в выборке ограниченного числа затравочных вершин в социальной сети для максимизации разброса информации (при числе вершин, которые в конечном итоге активируются в соответствии

с каскадной моделью).¹ Это число является случайной величиной, зависящей от исходов подбрасываний монеты в каскадной модели, и мы сосредоточимся на его математическом ожидании.² Формально обозначим через $X(S)$ (случайное) множество вершин, которые в конечном итоге активируются, когда вершины S выбираются в качестве заправочных, и определим *влияние* S как

$$f_{\text{inf}}(S) = \mathbf{E}[|X(S)|], \quad (20.10)$$

где математическое ожидание строится над случайными подбрасываниями монеты в каскадной модели. Влияние множества зависит как от графа, так и от активационной вероятности, причем большее число ребер или более высокая вероятность приводят к большему влиянию.

ЗАДАЧА: МАКСИМИЗАЦИЯ ВЛИЯНИЯ

Вход: ориентированный граф $G = (V, E)$, вероятность p и положительное целое число k .

Выход: выборка $S \subseteq V$ из k вершин с максимально возможным влиянием $f_{\text{inf}}(S)$ в каскадной модели с активационной вероятностью p .

Например, если вы раздаете k рекламных копий продукта и хотите выбрать получателей, чтобы максимизировать его принятие, то вы сталкиваетесь с задачей о максимизации влияния.

Задача 20.9 просит вас показать, что задача о максимальном охвате может рассматриваться как частный случай задачи о максимизации влияния. Поскольку частный случай является NP-трудным (задача 22.8), то и более общая

¹ Подробнее о задаче о максимизации влияния и ее многочисленных вариациях читайте в статье Дэвида Кемпе, Джона Клейнберга и Эвы Тардос «Максимизирование распространения влияния через социальную сеть» («*Maximizing the Spread of Influence Through a Social Network*», David Kempe, Jon Kleinberg, Éva Tardos, *Theory of Computing*, 2015).

² Математическое ожидание $\mathbf{E}[Y]$ случайной величины Y — это ее среднее значение, взвешенное соответствующими вероятностями. Например, если Y может принимать значения $\{0, 1, 2, \dots, n\}$, то $\mathbf{E}[Y] = \sum_{i=0}^n i \times \Pr[Y = i]$.

задача тоже. Может ли существовать быстрый и приближенно правильный эвристический алгоритм для задачи о максимизации влияния?

20.3.4. Жадный алгоритм

Задача о максимизации влияния напоминает задачу о максимальном охвате, где вершины играют роль подмножеств, а влияние (20.10) играет роль охвата (20.5). GreedyCoverage для последней задачи подходит для первой, если подставить новое определение (20.10) целевой функции.

GREEDYINFLUENCE

Вход: ориентированный граф $G = (V, E)$, вероятность $p \in [0, 1]$ и положительное целое число k .

Выход: множество $S \subseteq V$ из k вершин.

```

1  $S := \emptyset$  // выбранные вершины
2 for  $j = 1$  to  $k$  do // выбрать вершины одну за другой
   // жадно увеличивать влияние
   // (произвольно разрывать связи)
3    $v^* := \operatorname{argmax}_{v \in V} [f_{\text{inf}}(S \cup \{v\}) - f_{\text{inf}}(S)]$ 
4    $S := S \cup \{v^*\}$ 
5 return  $S$ 

```

ТЕСТОВОЕ ЗАДАНИЕ 20.6

Каково время выполнения простой реализации GreedyInfluence на графах с n вершинами и m ребрами? (Выбрать самый точный ответ.)

- а) $O(knm)$
- б) $O(knm^2)$
- в) $O(knm^{2m})$
- г) неясно

(Ответ и анализ решения см. в разделе 20.3.8.)
