참고문헌

2장

1 Wikipedia, https://en.wikipedia.org/wiki/Pseudocode
2 Wikipedia, https://en.wikipedia.org/wiki/Pseudocode#Syntax
3 구글 코드잼 2017년 통계, https://www.go-hero.net/jam/17/languages
4 안티 라크소넨, 『알고리즘 트레이닝』(인사이트, 2019), p2
5 카카오 신입 공채 1차 코딩 테스트 문제 해설: 카카오 테크 블로그, https://tech.kakao.com/2017/09/27/kakao-blind-recruitment-round-1/
6 Why Generics? - The Go Blog, https://blog.golang.org/why-generics

3장

1 Python has brought computer programming to a vast new audience, 〈The Economist〉, https://www.economist.com/science-and-technology/2018/07/19/python-has-brought-computer-programming-to-a-vast-new-audience
2 유튜브, https://youtu.be/J0Aq44Pze-w
3 유튜브, https://youtu.be/cKzP61Gjf00
4 Sharif, Bonita; Maletic, Jonathan I. (2010). "An Eye Tracking Study on camelCase and under_score Identifier Styles". 2010 IEEE 18th. pp. 196-205
5 윤인성, 『혼자 공부하는 파이썬』(한빛미디어, 2019), p133
6 http://google.github.io/styleguide/pyguide.html
7 http://google.github.io/styleguide/pyguide.html

4장

1 Wikipedia, https://en.wikipedia.org/wiki/Big_O_notation
2 문병로, 『쉽게 배우는 알고리즘』(한빛아카데미, 2018), p21
3 게일 라크만 맥도웰, 『코딩 인터뷰 완전 분석』(인사이트, 2017), p60
4 게일 라크만 맥도웰, 『코딩 인터뷰 완전 분석』(인사이트, 2017), p61
5 Wikipedia, https://en.wikipedia.org/wiki/Time_complexity
6 커트 건서로스, 『C++ 최적화』(한빛미디어, 2019), p137

7 대문자 O 표기법과 퀵 정렬의 시간 복잡도 - 구종만, http://theyearlyprophet.com/big-oh-time-complexity.html
8 Steven Skiena, 『Algorithm Design Manual』(Springer-Verlag, 2008), p33
9 Steven Skiena, 『Algorithm Design Manual』(Springer-Verlag, 2008), p35
10 대문자 O 표기법과 퀵 정렬의 시간 복잡도 - 구종만, http://theyearlyprophet.com/big-oh-time-complexity.html
11 Wikipedia, https://en.wikipedia.org/wiki/Amortized_analysis
12 커트 건서로스, 『C++ 최적화』(한빛미디어, 2019), p141
13 Wikipedia, https://en.wikipedia.org/wiki/Data_type)
14 "Is False == 0 and True == 1 an implementation detail or is it guaranteed by the language?" - Stack Overflow, https://stackoverflow.com/q/2764017/3513266
15 Python internals: Arbitrary-precision integer implementation, https://rushter.com/blog/python-integer-implementation/
16 Jon Steinhart, 『The Secret Life of Programs』(No Starch, 2019), p184
17 Why Python is Slow: Looking Under the Hood, https://jakevdp.github.io/blog/2014/05/09/why-python-is-slow/
18 Wikipedia, https://en.wikipedia.org/wiki/Data_structure
19 Wikipedia, https://en.wikipedia.org/wiki/Data_type
20 Wikipedia, https://en.wikipedia.org/wiki/Abstract_data_type

5장

1 Data Structures (Part I): Introduction - Python Like You Mean It, https://www.pythonlikeyoumeanit.com/Module2_EssentialsOfPython/DataStructures.html
2 Why Python is Slow: Looking Under the Hood - Pythonic Perambulations, https://jakevdp.github.io/blog/2014/05/09/why-python-is-slow/
3 Why Python is Slow: Looking Under the Hood - Pythonic Perambulations, https://jakevdp.github.io/blog/2014/05/09/why-python-is-slow/
4 Data Structures (Part II): Dictionaries - Python Like You Mean It, https://www.pythonlikeyoumeanit.com/Module2_EssentialsOfPython/DataStructures_II_Dictionaries.html
5 Python 3.6 Brings Better Dictionaries, Improved Async I/O, and More, https://www.infoq.com/news/2016/12/python-36-new-features/

6장

1 로버트 세지윅 외, 『알고리즘 개정4판』(길벗, 2018), p686
2 Python Reverse String - 5 Ways and the Best One - JournalDev, https://www.journaldev.com/23647/python-reverse-string
3 JDK-6804124 : (coll) Replace "modified mergesort" in java.util.Arrays.sort with timsort, https://bugs.java.com/bugdatabase/view_bug.do?bug_id=6804124
4 https://github.com/golang/go/issues/467#issuecomment-66049885
5 Interview with Tim Peters - YouTube, https://youtu.be/1wAOy88WxmY
6 Timsort - the fastest sorting algorithm you've never heard of, Medium, https://hackernoon.com/

timsort-the-fastest-sorting-algorithm-youve-never-heard-of-36b28417f399
7 RFC - 정보통신기술용어해설, http://www.ktword.co.kr/abbr_view.php?m_temp1=177
8 Jon Steinhart, 『The Secret Life of Programs』(No Starch, 2019), p25
9 How Python saves memory when storing strings, https://rushter.com/blog/python-strings-and-memory/

7장

1 Wikipedia, https://en.wikipedia.org/wiki/Array_data_structure
2 로버트 세지윅 외, 『파이썬을 이용한 컴퓨터과학 입문』(길벗, 2019), p493
3 게일 라크만 맥도웰, 『코딩 인터뷰 완전 분석』(인사이트, 2017), p767

8장

1 Wikipedia, https://en.wikipedia.org/wiki/Linked_list
2 Fast ring-buffer deque (double-ended queue), GitHub, https://github.com/gammazero/deque
3 Store reference to primitive type in Python? https://stackoverflow.com/q/16704377/3513266
4 Python Operator Precedence, https://www.mathcs.emory.edu/~valerie/courses/fall10/155/resources/op_precedence.html
5 What are the uses of a full-adder? - Quora, https://www.quora.com/What-are-the-uses-of-a-full-adder

9장

1 Wikipedia, https://en.wikipedia.org/wiki/Stack_(abstract_data_type)
2 조중필 외, 『알고리즘 문제 풀이 전략』(한빛미디어, 2016), p135
3 Wikipedia, https://en.wikipedia.org/wiki/Queue_(abstract_data_type)
4 토머스 코멘 외, 『Introduction to Algorithms』(한빛아카데미, 2014), p235
5 https://www.eecs.wsu.edu/~holder/courses/cse2320/lectures/l9/node12.html
6 https://www.eecs.wsu.edu/~holder/courses/cse2320/lectures/l9/node12.html

10장

1 Wikipedia, https://en.wikipedia.org/wiki/Double-ended_queue
2 Wikipedia, https://en.wikipedia.org/wiki/Priority_queue
3 What's the difference between heapq and PriorityQueue in python? https://stackoverflow.com/q/36991716/3513266
4 Python and the GIL https://boards.4channel.org/g/thread/64468331/python-and-the-gil

11장

1 Wikipedia, https://en.wikipedia.org/wiki/Hash_table
2 Wikipedia, https://en.wikipedia.org/wiki/Hash_function

3 Wikipedia, https://en.wikipedia.org/wiki/Hash_function
4 나라심하 카루만치, 『코딩 인터뷰 퀘스천』(영진닷컴, 2015), p462
5 Wikipedia, https://ko.wikipedia.org/wiki/생일_문제
6 Wikipedia, https://en.wikipedia.org/wiki/Four_color_theorem
7 Wikipedia, https://en.wikipedia.org/wiki/Birthday_problem
8 Wikipedia, https://speakerdeck.com/jakevdp/statistics-for-hackers
9 Wikipedia, https://en.wikipedia.org/wiki/Pigeonhole_principle
10 Wikipedia, https://en.wikipedia.org/wiki/Pigeonhole_principle
11 Wikipedia, https://en.wikipedia.org/wiki/Hash_table#Key_statistics
12 HashMap(Java SE 10 & JDK 10), https://docs.oracle.com/javase/10/docs/api/java/util/HashMap.html
13 https://en.wikipedia.org/wiki/Hash_function
14 Wikipedia, https://en.wikipedia.org/wiki/Hash_function#Hashing_integer_data_types
15 문병로, 『쉽게 배우는 알고리즘』(한빛아카데미, 2018), p225
16 https://bugs.java.com/bugdatabase/view_bug.do?bug_id=4045622
17 브라이언 커니핸, 데니스 리치, 『The C Programming Language』(Prentice Hall, 1988), p118
18 C++, Java 해시 테이블 구현, http://docs.likejazz.com/hash-table-implementations/
19 구글 브레인의 논문 「The Case for Learned Index Structures」, https://arxiv.org/abs/1712.01208
20 Java HashMap은 어떻게 동작하는가? https://d2.naver.com/helloworld/831311
21 나라심하 카루만치, 『코딩 인터뷰 퀘스천』(영진닷컴, 2015), p468
22 Objects/dictobject.c - CPython, https://github.com/python/cpython/blob/master/Objects/dictobject.c
23 찰스 페졸드 외, 『Beautiful Code』(한빛미디어, 2007), p298
24 Wikipedia, https://en.wikipedia.org/wiki/Hash_table
25 Asterisks in Python: what they are and how to use them - Trey Hunner, https://treyhunner.com/2018/10/asterisks-in-python-what-they-are-and-how-to-use-them/

12장

1 Wikipedia, https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)
2 박병하, 『수학의 감각』(행성B, 2018), p101
3 Historic Cities Research Project, https://visualhistory.livejournal.com/39249.html
4 MAA Euler Archive, https://www.maa.org/press/periodicals/convergence/leonard-eulers-solution-to-the-konigsberg-bridge-problem
5 Graph Theory and Data Science - Medium, https://towardsdatascience.com/graph-theory-and-data-science-ec95fe2f31d8
6 Wikipedia, https://en.wikipedia.org/wiki/Hamiltonian_path
7 The travelling salesman problem, https://www.sketchplanations.com/post/190464809359/the-travelling-salesman-problem-you-have-to-drop
8 위키백과, https://ko.wikipedia.org/wiki/NP_(복잡도)
9 Wikipedia, https://en.wikipedia.org/wiki/NP_(complexity)
10 xkcd, https://xkcd.com/399/
11 Wikipedia, https://en.wikipedia.org/wiki/Graph_traversal

12 Wikipedia, https://en.wikipedia.org/wiki/Depth-first_search#Pseudocode
13 Wikipedia, https://en.wikipedia.org/wiki/Depth-first_search#Pseudocode
14 Wikipedia, https://en.wikipedia.org/wiki/Breadth-first_search#Pseudocode
15 Wikipedia, https://en.wikipedia.org/wiki/Backtracking
16 문병로, 『쉽게 배우는 알고리즘』(한빛아카데미, 2018), p478
17 Backtracking, https://cs.lmu.edu/~ray/notes/backtracking/
18 Wikipedia, https://en.wikipedia.org/wiki/Constraint_satisfaction_problem
19 http://www.aistudy.co.kr/problem/constraint_satisfaction_problem.htm
20 In-depth Backtracking with LeetCode Problems - Part 1, https://medium.com/algorithms-and-leetcode/backtracking-e001561b9f28
21 박두순, 『MSE 이산수학』(한빛아카데미, 2019), p349
22 How to avoid "RuntimeError: dictionary changed size during iteration" error? https://stackoverflow.com/q/11941817/3513266

13장

1 Wikipedia, https://en.wikipedia.org/wiki/Shortest_path_problem
2 위키백과, https://ko.wikipedia.org/wiki/최단_경로_문제
3 로버트 세지윅 외, 『알고리즘 개정4판』(길벗, 2018), p630
4 구종만, 『알고리즘 문제 해결 전략 세트』(인사이트, 2012), p919
5 서은국, 『행복의 기원』(21세기북스, 2014), p183
6 데이비드 코펙, 『고전 컴퓨터 알고리즘 인 파이썬』(한빛미디어, 2019), p410
7 로버트 세지윅 외, 『알고리즘 개정4판』(길벗, 2018), p531
8 문병로, 『쉽게 배우는 알고리즘』(한빛아카데미, 2018), p334
9 토머스 코멘 외, 『Introduction to Algorithms』(한빛아카데미, 2014), p662
10 Wikipedia, https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm#Using_a_priority_queue
11 로버트 세지윅 외, 『알고리즘 개정4판』(길벗, 2018), p533

14장

1 Wikipedia, https://en.wikipedia.org/wiki/Tree_(data_structure)
2 조중필 외, 『알고리즘 문제 풀이 전략』(한빛미디어, 2016), p160
3 맥스 하웰의 트위터, https://twitter.com/mxcl/status/608682016205344768
4 토머스 코멘 외, 『Introduction to Algorithms』(한빛아카데미, 2014), p152
5 로버트 세지윅 외, 『알고리즘 개정4판』(길벗, 2018), p400
6 Wikipedia, https://en.wikipedia.org/wiki/Self-balancing_binary_search_tree
7 Wikipedia, https://en.wikipedia.org/wiki/Tree_traversal
8 Wikipedia, https://en.wikipedia.org/wiki/Tree_traversal
9 Wikipedia, https://en.wikipedia.org/wiki/Tree_traversal#Depth-first_search
10 Wikipedia, https://en.wikipedia.org/wiki/Tree_traversal#Depth-first_search

15장

1 Wikipedia, https://en.wikipedia.org/wiki/Heap_(data_structure)
2 Priority Queues with Binary Heaps - Practical Algorithms and Data Structures, https://bradfieldcs.com/algos/trees/priority-queues-with-binary-heaps/
3 토머스 코멘 외, 『Introduction to Algorithms』(한빛아카데미, 2014), p152
4 Priority Queues with Binary Heaps - Practical Algorithms and Data Structures, https://bradfieldcs.com/algos/trees/priority-queues-with-binary-heaps/
5 Priority Queues with Binary Heaps - Practical Algorithms and Data Structures, https://bradfieldcs.com/algos/trees/priority-queues-with-binary-heaps/
6 Wikipedia, https://en.wikipedia.org/wiki/Binary_heap#Extract
7 로버트 세지윅 외, 『알고리즘 개정4판』(길벗, 2018), p314

16장

1 Wikipedia, https://en.wikipedia.org/wiki/Trie

17장

1 Wikipedia, https://en.wikipedia.org/wiki/Sorting_algorithm
2 트위터, https://twitter.com/dhh/status/834146806594433025
3 유튜브, https://youtu.be/k4RRi_ntQc8
4 https://algorithmist.com/wiki/Bubble_sort
5 토머스 코멘 외, 『Introduction to Algorithms』(한빛아카데미, 2014), p40
6 Wikipedia, https://en.wikipedia.org/wiki/Merge_sort
7 Wikipedia, https://en.wikipedia.org/wiki/Quicksort#Lomuto_partition_scheme
8 Wikipedia, https://en.wikipedia.org/wiki/Quicksort#Lomuto_partition_scheme
9 Wikipedia, https://en.wikipedia.org/wiki/Sorting_algorithm#Stability
10 로버트 세지윅 외, 『알고리즘 개정4판』(길벗, 2018), p341
11 S. Skiena, 『Algorithm Design Manual』(Springer-Verlag, 2008), p125
12 Wikipedia, https://en.wikipedia.org/wiki/Insertion_sort
13 The Dutch flag - Holland, https://www.holland.com/global/tourism/information/general/the-dutch-flag.htm
14 Wikipedia, https://en.wikipedia.org/wiki/Dutch_national_flag_problem

18장

1 Wikipedia, https://en.wikipedia.org/wiki/Binary_search_algorithm
2 https://ai.googleblog.com/2006/06/extra-extra-read-all-about-it-nearly.html
3 파노스 루리다스, 『리얼월드 알고리즘』(길벗, 2019), p276, 2018
4 파노스 루리다스, 『리얼월드 알고리즘』(길벗, 2019), p282
5 https://docs.python.org/3.7/library/bisect.html#bisect.bisect_left

19장

1 Jon Steinhart, 『The Secret Life of Programs』(No Starch, 2019), p4
2 Wikipedia, https://en.wikipedia.org/wiki/XOR_gate
3 What is the full adder? Quora, https://www.quora.com/What-is-the-full-adder/ 중 Abhishek Singh의 답변
4 Jon Steinhart, 『The Secret Life of Programs』(No Starch, 2019), p25

20장

1 위키백과, https://ko.wikipedia.org/wiki/슬라이딩_윈도

21장

1 Wikipedia, https://en.wikipedia.org/wiki/Greedy_algorithm
2 문병로, 『쉽게 배우는 알고리즘』(한빛아카데미, 2018), p367
3 Wikipedia, https://en.wikipedia.org/wiki/Greedy_algorithm#Specifics
4 Wikipedia, https://en.wikipedia.org/wiki/Knapsack_problem
5 Wikipedia, https://en.wikipedia.org/wiki/Greedy_algorithm

22장

1 Wikipedia, https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm
2 Wikipedia, https://en.wikipedia.org/wiki/Merge_sort
3 Divide and conquer algorithms - Khan Academy, https://www.khanacademy.org/computing/computer-science/algorithms/merge-sort/a/divide-and-conquer-algorithms
4 조중필 외, 『알고리즘 문제 풀이 전략』(한빛미디어, 2016), p470

23장

1 Wikipedia, https://en.wikipedia.org/wiki/Overlapping_subproblems
2 Dynamic Programming vs Divide-and-Conquer Or Divide-and-Conquer on Steroids, https://itnext.io/dynamic-programming-vs-divide-and-conquer-2fea680becbe
3 알렉산더 스테파노프, 『알고리즘 산책』(길벗, 2018), p176
4 Fibonacci number - Wikipedia, the free encyclopedia https://en.wikipedia.org/wiki/Fibonacci_number
5 https://www.ics.uci.edu/~eppstein/161/960109.html
6 Memoization - Interview Cake, https://www.interviewcake.com/concept/java/memoization
7 Memoization - Interview Cake, https://www.interviewcake.com/concept/java/memoization
8 알렉산더 스테파노프, 『알고리즘 산책』(길벗, 2018), p178
9 Wikipedia, https://en.wikipedia.org/wiki/Knapsack_problem
10 Are dictionaries ordered in Python 3.6+?, https://stackoverflow.com/q/39980323/3513266

부록 A

1 ['토스(Toss)'에 퇴사자가 많은 이유는?] '팀내 평가로 퇴사 권고' 분위기 속 매달 10명씩 사표 - 중앙일보, https://jmagazine.joins.com/economist/view/329794
2 현대차 · KT 이어 LG도 수시채용… 대기업 공채제도 사라진다 - 파이낸셜 뉴스, https://www.fnnews.com/news/202006091735293353
3 존 손메즈, 『커리어 스킬』(길벗, 2019), p216
4 폴 아담스, 『Grouped 세상을 연결하는 관계의 비밀』(에이콘출판, 2012)
5 에드워드 글레이저, 『도시의 승리』(해냄출판사, 2011), p100
6 Detroit Tops 2013 List Of America's Most Miserable Cities, 「Forbes」, https://www.forbes.com/sites/kurtbadenhausen/2013/02/21/detroit-tops-2013-list-of-americas-most-miserable-cities/
7 Wikipedia, https://en.wikipedia.org/wiki/All_your_base_are_belong_to_us
8 니시노 류타로, 『IT 개발자의 영어 필살기』(책만, 2020), p6 추천사
9 존 손메즈, 『커리어 스킬』(길벗, 2019), p184
10 존 손메즈, 『커리어 스킬』(길벗, 2019), p164
11 삼성그룹 26년 인사담당 "면접장 들어서는 순간 당락 80% 결정" - 조선일보, http://news.chosun.com/misaeng/site/data/html_dir/2016/04/21/2016042102546.html
12 존 손메즈, 『커리어 스킬』(길벗, 2019), p194
13 존 손메즈, 『커리어 스킬』(길벗, 2019), p201
14 존 손메즈, 『커리어 스킬』(길벗, 2019), p202
15 Judges are more lenient after taking a break, study finds - The Guardian, https://www.theguardian.com/law/2011/apr/11/judges-lenient-break
16 Russel Cohen, An Analysis of Hash Map Implementations in Popular Languages https://rcoh.me/posts/hash-map-analysis/
17 「이데일리」, 2015/07/14, https://news.v.daum.net/v/20150714152046530
18 토스, "업계 최고 능력자 모셔라" 파격 보상안 발표, https://blog.toss.im/2019/10/31/newsroom/press-release/toss-recruit-withsob/

부록 B

1 카카오 신입 공채 1차 코딩 테스트 문제 해설, https://tech.kakao.com/2017/09/27/kakao-blind-recruitment-round-1/