

# Expert 과정소개

Hancom Education Co. Ltd.

JooHyun - Lee (comkiwer)

## Expert 과정 도입 배경

Expert 인증은 Pro 인증자 내에서 선발합니다.

그러므로, Pro 개발자가 갖춰야 할 기본기는 필수가 될 것이고, 더해서

**"빠르고 정확한 Implementation"**을

주요 허들로 설정합니다.

성능 이전에, Expert 문제에서 제시하는 요구사항을 만족하는 결과물을

**최대 2시간 내**에 얻지 못하면 인증 가능성은 매우 낮다고 볼 수 있습니다.

나머지 시간에 과제경험을 통해 축적된 창의적 아이디어로 최적화를 달성하셔야 하겠습니다.

- 20151120 Expert를 준비하시는 분들께 -

## Expert 과정 도입 배경

“현업에서 스크래치 위주의 개발을 5 ~ 10년간 꾸준히 수행해 온 능력있는 (Native Speaker) 개발자를 뽑는다”

- Native Speaker라면 통상적인 일상 생활, 각종 업무 처리 등에 있어 상대방이 의도하는 바를 파악하고, 또한 자신이 의도하는 바를 표현함에 있어 쉽고 빠르고 정확하여 막힘이 없는 사람을 의미하는 것

참조 : 20160810 8월 expert 시험과 관련하여

## Expert 과정 도입 배경

사실 Expert 출제는 고차원 Legacy 알고리즘을 선택하고 그것을 포장하는 방식으로 하지 않습니다.

저의 과제 경험과 SW 사고를 바탕으로,  
특정 도메인/레이어 개발자에게 치우치지 않으면서도,  
Programming Skill이라는 공통적인 역량을 측정할 수 있는  
일반화된 문제를 내려는 노력을 하고 있습니다.

기본 출제 아이디어를 도출한 후,  
전문가들의 리뷰와 손질을 거쳐 최종적으로 한 문제가 나옵니다.

매 차수에 따라 응시자 분들의 성향이나 분야에 따라 호불호가 있다는 점 잘 보고 있으며,  
이를 최소화하기 위한 노력을 지속적으로 하고 있습니다.

-20160810 8월 expert 시험과 관련하여. -

## Expert 양성될 수 있는가?

Expert 인증은 100% pull 정책입니다. 어떠한 사내 제도와도 연결되지 않습니다.

(PRO 인증은 Architect 양성이나 Best Reviewer 양성의 입과 조건인 것과 비교)

따라서, Expert Programmer 인증은 우수 개발자들이 선의의 경쟁으로 실력을 겨룬다는 측면에서, 삼성그룹 내 S/W 개발 문화이자 S/W engineer의 vision 대상 중 하나입니다.

이러한 의미로, 회사는 Expert 인증을 위한 어떠한 양성 프로그램도 제공하지 않으며, S/W engineer 스스로 학습하고 도전하는 장치로써 적극 지원을 할 뿐입니다. (Knox ID 표기 등)

계속된 연습을 통해 도달할 수 있는 수준인지의 질문에 대한 답은 이렇습니다.

"인간 잠재력의 끝은 누구도 알 수 없다. 끊임없는 훈련과 도전으로 스스로의 한계를 확인하는 길 뿐"

[20210401 Ex는 양성될 수 있는가에 대한 선배님들의 생각이 궁금합니다.]

# Pro 시험과의 비교

문제

	pro	expert
문제	자세한 문제 설명 및 예제 설명	문제 설명 거의 없음. 코드를 보고 파악 해야함. 23년 7월부터는 문제 설명이 늘어남.
요구사항	목표 성능의 한계를 극복할 수 있는 체계적, 논리적, 알고리즘적 접근이 가능한지를 보는 것.	오랜 실무 프로그래밍 경험을 통해 다양한 형태의 문제를 창의적 아이디어와 빠르고 정확한 솔루션으로 해결할 수 있는 축적된 스킬을 보는 것.
채점방법	TC50개 제한 시간안에 모두 수행 & 답이 모두(가끔은 거의) 맞음 (TC 1개에 2점 2 * 50 = 100점)	요구사항을 만족하는 답안에 대해 성능이 높을수록 높은 점수를 부여. (많은 경험이 있는 참가자가 아니면 bigO 개념이 아니기 때문에 솔루션의 boundary 예측 조차 어려울 수 있음.)
입력	파일로부터 입력	random으로 생성 또는 파일 입력(23년7월, 24년2월)
합격률	평균 4~5%	0.5%내외
재인증율	20% 이하	10%(2016년 25%)
제도	PRO 인증은 Architect 양성이나 Best Reviewer 양성의 입과 조건	Expert 인증은 100% pull 정책. 어떠한 사내 제도와도 연결되지 않음.

	2023년 5월까지 : J(서장석)님	2023년 7월부터 : new출제진
최근 문제	... 220903 TextEditor8 221112 PUZZLE2 230204 VRAM 230318 BANK 230513 Compression3	230729 로봇청소기4(input) 231125 물품배송 240330 Taxi Assignment 240706 우주선착륙 241116 교차로신호제어 250329 화물기차 250719 노후화된 도로  230916 안테나 배정 240203 IoT Appliance(input) 240525 Germs 240907 드론배송 250208 ProteinFold 250517 하르방 250913 ChipDesign
입력 파일	없음	있는 경우가 현재까지 2문제
문제 설명	거의 없음	이전보다는 자세한 편
문제 유형	대부분이 <b>NP (근사해 : Heuristic)</b>	<b>NP (근사해 : Heuristic) 유형의 문제</b> 해결을 위한 핵심 아이디어를 묻는다는 점에는 변화가 없지만 좀 더 기획되고 복잡해짐.
main소스	대체로 길지 않은 편	긴 편 : 문제 읽기, 이해, 분석에 시간이 걸린다.
구현양	짧은 편	긴 편 : 구현력이 중요해짐
채점	90% 근사해 + 성능(실행 속도) 프로그램이 실행 속도가 가능한 빨라야 한다.	100% 근사해 (실행 속도는 제한 시간만 만족하면 된다.) 제한 시간 안에서 문제에서 요구하는 점수가 가능한 좋아야 한다.
리더보드	상위 3명의 점수를 보여줌. (컷 점수 추측 가능)	더 이상 제공하지 않음.(컷 점수를 추측하기 힘들다.)

- Expert 문제는 NP문제이다.

거의 모든 문제에서 최적해를 보장하는 Problem Solving Contest 문제와 다른 접근이 필요하다.

제한시간(10초) 안에 최적해를 구할 수 없다. (일부 seed 또는 일부 TC에서 최적해는 가능하나 전체는 불가능하다.) 따라서 Heuristics을 이용한 근사해를 구해야 한다.

## ➤ Heuristics

휴리스틱(heuristics) 또는 발견법(發見法)이란 불충분한 시간이나 정보로 인하여 합리적인 판단을 할 수 없거나, 체계적이면서 합리적인 판단이 굳이 필요하지 않은 상황에서 문제 해결에 빠르게 사용할 수 있게 보다 쉽게 구성된 간편추론의 방법. – wiki

휴리스틱의 어원은 라틴어의 "heuristicus" 와 그리스어 "heuriskein"에서 유래하였으며, "찾아내다"(find out)와 "발견하다"(discover)를 뜻한다

## ➤ 레거시 하거나 특정 도메인에 치우친 문제는 거의 출제되지 않는다.

## • 문제 해결 방법

- Pro과정의 기본기는 필수 + 경험 + 직관과 통찰력(**heuristics**) + 구현력 이 필요하다.
- 대체로 Gready한 해결책(**heuristics**) 을 기반으로 문제에서 요구하는 최적해에 가까운 근사해를 구한다.
- 문제해결 방법과 구현이 쉽지 않은 문제가 간혹 있지만, 상당히 많은 문제해결에서 뭔가 복잡하고 지나치게 어려운 즉 4시간 안에 구현되기 어려운 방법을 구상했다면 다시 생각해봐야 한다.

- 1종 오류 (Type I error):** 귀무가설( $H_0$ )가 실제로는 참이어서 채택해야 함에도 불구하고 표본의 오차때문에 이를 채택하지 않는 오류를 말합니다. 보통  $\alpha$  (알파)로 표기하고 유의수준이라고 부릅니다.
- 2종 오류 (Type II error):** 귀무가설( $H_0$ )가 거짓이라서 채택하지 말아야 하는데 표본의 오차 때문에 이를 채택하는 오류를 말합니다. 보통  $\beta$  (베타)로 표기합니다.

통계적 결정		실제상황	$H_0$ 가 사실 (참)	$H_0$ 가 허위 (거짓)
			옳은결정 확률 = $1 - \alpha$	제II종오류 확률 = $\beta$
			제I종오류 확률 = $\alpha$	옳은결정 확률 = $1 - \beta$
				검정력

유의수준

Expert시험은 Expert 기준에 미달하는 수험자가 합격하는 것을 최소화 하는 것이 1차 목적으로,  
따라서 Expert 수준의 실력자임에도 불합격 될 수 있다.

# Expert 대비는 어떻게?

문제

[ 20210408 expert는 어떤식으로 공부해야 할까요 ]

돌\*\*\* =====

저는 입사 후 solving talk 에 있는 Expert 후기들을 모두 정독하여  
거기서 공부 방법 및 노하우를 얻어서 어려운 알고리즘 지식 없이도 빠르게 합격할 수 있었습니다.  
또한 인증 이후에도 한동안 지속적으로 Expert 시험을 봐서 추가로 PASS 를 받았습니다.

...

그 방법은 기출 문제를 분석하고 합격 커트라인에 필요한 최적화 방법을  
사용된 테크닉들을 뼈속까지 익히는 것이었습니다.

계속 새로운 문제가 등장해도 합격에 도달하는 핵심 접근법과  
최적화 및 풀이법은 정해진 범위에서 돌고 돌더라고요.

이를 위해 가장 중요한 것은,

1. 기출 문제를 복기 할 것
2. 실제 인증 받은 점수를 재현해내는 것 이었습니다.

보통의 사람들은 2 번을 해내지 못해서 인증 받지 못하는 코드로 공부를 하기 때문에  
다음에 유사한 문제가 나와도 지속적으로 떨어진다는 점을 발견했구요.

...

대략적으로 2 년치의 핵심 기출 문제를 풀이하면서 기초를 익히면서,  
남들보다 조금 더 앞서가서 리더보드에 드는 노하우를 공유하는 것이었습니다.

...

# Expert 대비는 어떻게?

문제

[ 20210408 expert는 어떤식으로 공부해야 할까요 ]

] =====

"계속 새로운 문제가 등장해도 합격에 도달하는 핵심 접근법과 최적화 및 풀이법은 정해진 범위"

범위를 정해서 출제를 하는게 아니라,

아시다시피 SW라는게 달나라 가는 핵심 기술이란게 없으니까요.

어느 정도 skill set이 쌓이면 대부분 problem은 거기서 거깁니다.

현업에서 부딪히는 문제도 "정해진 범위"를 벗어나는 경우가 별로 없습니다.

그런 취지의 제도입니다.

참고로 Expert 출제는, 지난 30년 현업 개발을 하면서

직접 부딪힌 problem 기억들을 되살리고 일반화 해서 **외부 협업 없이 직접 출제**하는 것입니다.

클린 아키텍쳐를 쓴 로버트 마틴 얘기처럼 30년 전이나 지금이나 SW skill은 별로 달라진게 없거든요.

이 의미는, 여러분이 지금 갖고 닦는 skill이 30년 뒤에도 위력을 발휘할 개연성이 높다는 것입니다.

노이만 형이 고안한 프로그램 내장방식 micro processor가 계속 쓰이는 때 까지는..

# Expert 대비는 어떻게?

문제

[20210408 expert는 어떤식으로 공부해야 할까요]

글\*\*\* =====

괴물분들에게는 범접하지 못하는 익린이로써 제가 준비했던 방법을 공유드리면..  
기출문제를 푼다. 문제가 없으면 몸으로 때운다(직접 시험쳐서 문제를 가져온다) 였습니다.  
다만 문제를 풀때는 시험장에서 1등 점수보다 조금이라도 높은 솔루션을 찾아보려고  
오래(길게는 한달이상) 풀었습니다.

R\*\*\* =====

말씀하신대로 기출 문제를 찾기가 어려워서 직접 시험 보고 복기한 문제를 소중히 여기면서 두번 세번 풀었습니다.  
주변에 익스 준비하시는 분들 보면 지금까지 본 시험들 다 복기 하셨을 거라 익스 합격하신 분들이라면  
그 당시 기출문제들 많이 가지고 있을 수도 있습니다.

R\*\*\*\*\* =====

한, 두방에 수월하게 따는 소수의 분들을 제외하면 많은 분들이 비슷한 것 같네요.  
시험보고나서 집에서 마저 풀어보기. 최적화가 목적인거라면 **극한의 극한까지 짜내서 최적화** 해보기  
(저는 혼자 끙끙댔지만, 조언 구할 분 있으면 더 좋을 듯)  
그리고 나서 시험장에서 왜 풀지 못했을까 자가분석 -> 분석 결과로 개선 방안 찾기 -> 실천  
(부족했던 점이 뭔지에 따라 뭘 하는지 종류가 조금 달라지지만 **결국은 공부와 연습의 반복**)

일정

## 일정 계획

- 2주 10일간 진행.
- ✓ 커리큘럼:  
학습 page([www.codepass.co.kr](http://www.codepass.co.kr))의  
[SDC] 25년 2차 Expert 대비 과정 참조
  
- 일과 시간 : 08:00 ~ 17:00  
쉬는 시간 : 매시간마다 10~20분  
점심 시간 : 11:40 ~ 13:00

# 코딩 스타일

1. IDE : VS2017, ( 보조 : codeblocks )
2. 확장자 : \*\*\*.cpp 를 권장. (c유저 일지라도 확장자는 .cpp로)
3. c의 장점 (scanf, printf : 속도때문에), cpp의 장점(참조자, 클래스 멤버함수 등) 을 혼용.
4. 정적 배열의 크기는 가능하다면 조금 여유 있게 설정.
5. 식별자(변수 및 함수 이름)는 혼동의 여지가 없다면 비교적 간략하게,  
필요하다면 의미를 담아 .
6. ClassName은 대문자로 시작,  
camel Notation과 snake Notation을 병용
7. 배열의 indexing은 상황에 따라 zero base와 one base를 혼용.
8. #define LM 1000 => constexpr int LM = 100;
9. 매크로 함수는 지양하고 inline 키워드를 사용하는 것으로...
10. stl을 사용할 수 있다면 문제 분석 및 문제 풀이 초기의 스케치 코드에 이용할 수 있음.

## 당부의 말씀

1. 과정을 수강하는 동안은 학습하는 것을 우선순위로 놓기.
2. 여력이 된다면 문제를 미리 풀어보기.  
힘들다면 문제를 미리 읽어보는 것 만이라도...
3. 강의내용을 그냥 받아들이기 보다는 비판적으로 수강 ...  
질문 및 오타 등 지적할 내용이 있다면 눈치 보지 말고 바로 바로 ...
4. Expert 시험은 main.cpp, user.cpp 두 개의 파일을 이용함. 반드시 두개의 파일로 문제 풀기.
5. 타인의 코드를 분석해 보는 것도 실력향상이 많은 도움이 됨.  
( 내꺼는 원래 내꺼 남의 것도 내꺼 ^~^ )

\* 예 : software academy 공개된 문제의 제출된 코드를 참고하기

부록 : pro 기출목록

23년 6월 25 ~ 25년 11월8일까지 199문제

# 2021년 pro 기출 목록

: 6월 25부터

num	date	SWEA title	result	Algorithm, Data-Structure : C++
1	20210625	[H2116]조각맞추기게임 - STL	DS : 0.3%	A_Hash(DA), array
2	20210702	[H2118]병사관리	DS : 6.0%	LinkedList, list(splice)
3	20210703	[H2119]출석체크	DS : 2.4%	string parsing, sort, LL
4	20210709	[H2120]파일손상	DS : 12.1%	LinkedList, Hash : Tree
5	20210710	[H2121]놀이공원	DS : 6.6%	Heap(set, map), Hash : time flow
6	20210717	[H2122]자전거 대여소	DS : 16.0%	Hash, Heap:PQ(set)
7	20210723	[H2123]롤러코스터 설계	DS : 2.0%	deque, 2stack, (A_Hash)
8	20210807	[H2125]조립식 컴퓨터	DS : 0.2%	implement, (data compressing, A_hash), rotate
9	20210813	[H2126]클라우드 스토리지	DS : 1.5%	tree + reverse_track, (unordered_)set/map
10	20210814	[H2127]조직관리	DS : 3.7%	tree, (unordered_)set, (unordered_)map, sort
11	20210820	[H2128]시간여행자의 주식	DS : 7.1%	Heap:PQ(set)
12	20210821	[H2129]파일 캐시	DS : 11.9%	hash, DLL(map)
13	20210827	[H2131]암호 장비(Py)	DS : 3.5%	stack, all case search(dfs, bfs, hard_cording)
14	20210903	[H2132]AI 로봇	DS : 5.9%	Heap:PQ(set)+vector or PQ+DLL , IDEA, timeflow
15	20210910	[H2133]앱평가	DS : 5.3%	Heap:PQ(set)
16	20210911	[H2134]순위표	DS : 5.4%	insertionSort + binarySearch
17	20210917	[H2135]병렬 컴퓨팅	DS : 0.7%	Heap:PQ(map) + timeflow
18	20210925	[H2136]문자열 관리 프로그램(py)	DS : 2.3%	Hash(DA) + idea
19	20211001	[H2138]중고시장 앱	DS : 2.1%	A_hash
20	20211002	[H2139]파일 다운로드	DS : 3.3%	Hash, Graph(bfs), levelTable

num	date	SWEA title	result	Algorithm, Data-Structure : C++
21	20211008	[H2140]리스트 복사	DS : 0.5%	hash + reverse_track
22	20211016	[H2141]카메라 오류 보정	DS : 1.8%	A_hash, chaining, rotate
23	20211022	[H2143]생산 시스템(Py)	DS : 1.4%	hash + list(queue), (pq, set, map)
24	20211029	[H2144]동선 추적	DS : 6.1%	vector(list), list, implement
25	20211105	[H2145]지형도	DS : 2.2%	array, implement, insertionSort
26	20211112	[H2146]회의 예약 시스템	DS : 1.1%	hash, check Array(set)
27	20211113	[H2147]배달시스템(Py)	DS : 2.9%	hash, PQ(heap)
28	20211120	[H2148]악보	DS : 4.7%	Graph, Topological_Sort
29	20211126	[H2150]메모장 프로그램	DS : 5.0%	implementation, list(string, deque)
30	20211203	[H2151]Process Scheduler	DS : 0.3%	time flow, queue
31	20211204	[H2152]식당의 가치(Py)	DS : 10.8%	hash, list(array)
32	20211210	[H2153]입자 시뮬레이션	DS : 1.2%	segment compressing:chaining(vector)
33	20211211	[H2154]이미지 검색하기	DS : 4.8%	bitwise compressing:chaining(vector), trie:DFS
34	20211218	[H2155]P2P 네트워크	DS : 1.7%	BFS or DFS : vector, list...
35				
36				
37				
38				
39				
40				

# 2022년 pro 기출 목록

num	date	SWEA title	result	Algorithm, Data-Structure : C++
1	20220114	[H2202]워드 프로세서(Py)	DS : 1.5%	string, hash
2	20220115	[H2203]배달 라이더	DS : 2.9%	pq, time flow
3	20220121	[H2204]딱지 게임	DS : 0.8%	union find
4	20220122	[H2205]단어검색	DS : 3.1%	string, trie
5	20220128	[H2207]bingo	DS : 1.0%	a-hash, vector chaining
6	20220211	[H2208]그룹 메세지	DS : 0%	ad_hoc
7	20220212	[H2209]순서맞추기게임(Py)	DS : 4.8%	idea, Bsearch
8	20220218	[H2210]나무를 지키는 경찰	DS : 6.7%	pq, time flow, bfs(dfs)
9	20220219	[H2211]인터프리터 언어 IDE	DS : 0%	Stack, list
10	20220225	[H2213]계산 게임	DS : 4.5%	deque
11	20220304	[H2214]메모리 시스템(Py)	DS : 9.3%	adhoc, list : bst
12	20220305	[H2215]라인트리스(Linetris)	DS : 0%	adhoc, bfs(dfs)
13	20220311	[H2216]긴 사다리 게임	DS : 7.1%	map, upper_bound
14	20220319	[H2217]끝말잇기2	DS : 13.4%	string, pq, list, hash(trie)
15	20220325	[H2218]집 만들기	DS : 0%	BFS(DFS), adhoc
16	20220401	[H2219]우주자원개발	DS : 0%	hash, prefixSum_2D
17	20220402	[H2220]성적조회	DS : 6.7%	set(BST), unordered_map(hash), bbst, segTree, (index)
18	20220408	[H2222]미생물	DS : 0%	pq, time flow, hash(or simple search), (index)
19	20220416	[H2223]문명	DS : 3.6%	union find, hash
20	20220422	[H2224]토너먼트(Py)	DS : 6.7%	find recursive rule

num	date	SWEA title	result	Algorithm, Data-Structure : C++
21	20220506	[H2226]지하철 노선 관리	DS : 2.5%	BFS, Floyd
22	20220520	[H2227]삼국지게임	DS : 3.3%	union find, hash
23	20220521	[H2228]막대놓기(Py)	DS : 5.7%	BFS, (area search)
24	20220527	[H2230]가게관리	DS : 9.1%	hash, PQ(heap)
25	20220528	[H2231]TextEditor 7	DS : 0%	list, string
26	20220610	[H2232]섬지키기(Py)	DS : 3.3%	A-hash(vector chaning), BFS
27	20220611	[H2233]죄인의숲	DS : 4.3%	BFS+Hash
28	20220617	[H2234]주가변동	DS : 2.0%	sqrt decomposition
29	20220624	[H2235]생존열차	DS : 13.5%	PQ, (index)
30	20220625	[H2236]DNASequene	DS : 6.4%	Trie, Hash, map
31	20220702	[H2237]세균번식	DS : 4.8%	PQ+BFS(DFS)
32	20220708	[H2239]주차장	DS : 11.7%	PQ[set] + hash(renum), time flow
33	20220715	[H2240]로그인대기열	DS : 5.6%	Hash(vector chaining)
34	20220723	[H2241]호텔검색서비스(py)	DS : 10.9%	set + vector
35	20220729	[H2242]얼음행성탐사	DS : 1.3%	BFS + vector[set] + (lower_bound)
36	20220805	[H2244]무질서한하노이탑(py)	DS : 2%	recursive rule, sort, (stack)
37	20220806	[H2245]모든패어찾기	DS : 7.3%	idea, grouping, Bsearch
38	20220812	[H2246]마법문자열	DS : 5.5%	list(splice), time flow
39	20220820	[H2247]생산기계	DS : 10.1%	set(PQ), hash, Bsearch
40	20220826	[H2248]회의관리	DS : 3.7%	set[sampleSol], vector(LL), hash

num	date	SWEA title	result	Algorithm, Data-Structure : C++
41	20220902	[H2249]평면의 조각	DS : 6.7%	sort, idea(Geo), Bsearch
42	20220916	[H2251]쏟아진블록(py)	DS : 14.2%	naive구현, (hash), BFS(DFS)
43	20220917	[H2252]큰퍼즐	DS : 0.8%	hash
44	20220923	[H2253]좌석배정	DS : 3.1%	UnionFind 응용
45	20220924	[H2254]개발협업도구	DS : 0.7%	hash, dfs, adjancentArray
46	20221001	[H2255]물류허브(py)	DS : 1.9%	Dijkstra, hash(renum)
47	20221007	[H2257]수식계산기	DS : 0%	stack, hash
48	20221008	[H2258]일정관리4	DS : 13.9%	map(set), hash
49	20221014	[H2259]다운로드서비스	DS : 1.2%	PQ(set), time flow naive
50	20221028	[H2260]인플루언서	DS : 5.6%	BFS, set(pq, vector+nth_element)
51	20221104	[H2262]주거지검색서비스	DS : 1.2%	Dijkstra, set(PQ), (hash)
52	20221105	[H2263]인터프리터IDE2	DS : 3.2%	stack, list
53	20221111	[H2264]사물함배정	DS : 10.4%	PQ, set, hash
54	20221119	[H2265]조별경기	DS : 7.1%	union find
55	20221125	[H2266]자유무역협정(py)	DS : 3.3%	union find, hash
56	20221209	[H2269]항공노선관리	DS : 5.9%	Dijkstra(N*N)
57	20221210	[H2270]최대최소값 찾기	DS : 0%	seg tree, sqrt decomp
58	20221216	[H2271]좌석예약	DS : 3.9%	pq, bfs
59	20221217	[H2272]직선왕국	DS : 0%	set, pq
60	20221202	[H2268]블록제거게임	DS : 1.7%	list

# 2023년 pro기출 목록

num	date	SWEA title	result	Algorithm, Data-Structure : C++
1	20230114	[H2302]XY Polynomial(py)	DS : 5.3%	renum_hash, calc polynomial
2	20230127	[H2304]기지국설치	DS : 4.2%	bisearch, hash, (set,map:sort)
3	20230128	[H2305]즉석계산기	DS : 7.9%	stack, calc, precalc_bucket(idea)
4	20230211	[H2306]우산창고	DS : 7.5%	pq(set), graph(tree search)
5	20230217	[H2307]개체 관리하기	DS : 2.1%	vector, list
6	20230218	[H2308]꿀벌 집	DS : 0.7%	idea, Ahash
7	20230224	[H2309]출근길 라디오(py)	DS : 11%	SqrtDecomp, segTree, Fenwick
8	20230304	[H2311]학교배정(py)	DS : 1.6%	hash, set
9	20230310	[H2312]검색엔진2	DS : 0.6%	hash(trie), partial, nth
10	20230324	[H2313]계통수	DS : 11.3%	renum_Hash, LCA, idea
11	20230401	[H2315]사각형관리	DS : 2.1%	list, decomp(grouping)
12	20230414	[H2316]뉴스알림	DS : 4.6%	pq, vector, time_base
13	20230429	[H2318]블록쌓기게임	DS : 1.3%	sqrtDecomp, lazySegTree
14	20230512	[H2319]공지방송	DS : 4.7%	set,map,pq, Functor
15	20230520	[H2320]미생물가계도(py)	DS : 6.6%	hash, lca, sqrtDeomp
16	20230526	[H2321]단어게임	DS : 5.2%	hash, map, imple
17	20230602	[H2323]로봇청소기3(py)	DS : 1.1%	bfs, dfs
18	20230617	[H2324]호텔방문	DS : 2.4%	ShortestPath(Dijkstra:ElogV)
19	20230624	[H2325]단어미로	DS : 9.7%	Hash, Trie
20	20230707	[H2327]조직개편	DS : 0.8%	Hash, dfs(bfs)

num	date	SWEA title	result	Algorithm, Data-Structure : C++
21	20230714	[H2328]전자상가	DS : 2.0%	BinarySearch
22	20230722	[H2329]전자게시판	DS : 10.8%	pq, hash
23	20230805	[H2331]상품권배분	DS : 19.2%	BinarySearch, dfs(bfs)
24	20230811	[H2332]로봇운송차단	DS : 2.2%	AHash:vectorChaining, bfs
25	20230819	[H2333]박테리아 번식	DS : 3.0%	pq, BFS
26	20230901	[H23]전기차대여소	DS : 12.2%	BFS, ShortestPath
27	20230908	[H2336]물품보관	DS : 1.6%	sqrtDecomp[segTree] / set,map
28	20230923	[H2337]승강제리그	DS : 7.8%	set, indexPQ
29	20231007	[H2339]파일저장소	DS : 11.1%	set(map), (pq), hash
30	20231013	[H2340]LineEditor	DS : 0%	list, string : [CordingHard]
31	20231104	[H2342]Push	DS : 15.9%	BFS
32	20231110	[H2343]KNN	DS : 10.7%	Hash
33	20231118	[H2344]숫자조각게임	DS : 12%	Hash
34	20231124	[H2345]비용과 시간	DS : 18.7%)	Dijkstra, DP
35	20231202	[H2346]감시탑	DS : 9.6%	SqrtDecomp+vector, set, map
36	20231208	[H2348]국가행정	DS : 7.5%	PQ(set) + Bisearch
37	20231216	[H2349]신소재케이블2	DS : 16.6%	bfs, dfs, lca
38	20231223	[H2350]이미지 편집기	DS : 2.6%	bfs, dfs, pq(Huffman), bitwise

# 2024년 pro기출 목록

num	date	SWEA title	result	Algorithm, Data-Structure : C++
1	20240113	[H2402]화물운송	DS : 16.2%	BFS+biSearch
2	20240119	[H2403]단어암기장	DS : 3.4%	<b>sqrtDecomp</b> + list
3	20240131	[H2404]구슬치기	DS : 0%	implement(grouping)
4	20240202	[H2405]OnlineStreamer	DS : 1.4%	<i>sqrtDecomp</i> : <b>segtree</b>
5	20240217	[H2407]상징물 설치2(py)	DS : 4.7% c(0)	Hash(grouping)
6	20240223	[H2409]예약버스	DS : 1.1%	Dijkstara(spfa)
7	20240309	[H2410]던전탈출	DS : 7.7%	BFS + Dijkstra(spfa)
8	20240316	[H2411]네모 타일(py)	DS : 7.5% c(1),p(10)	Hash
9	20240322	[H2412]마을	DS : 0%	SqrtDecom(grouping), hash, UnionFind, list_splice
10	20240406	[H2413]카페	DS : 2%	Hash, Queue
11	20240412	[H2415]광물합성	DS : 3.3%	BinarySearch, [H2328]전자상가
12	20240419	[H2416]전송시간	DS : 1.7%	Dijkstra, Opt
13	20240503	[H2418]사진메타데이터관리	DS : 0%	Queue, Hash
14	20240518	[H2419]배송로봇	DS : 2.1%	Floyd, Dfs(permutation)
15	20240524	[H2420]그림판2	DS : 3.9%	D&C + BFS
16	20240608	[H2423]지하도시 건설	DS : 9.5%	NaiveSearch[Map, Set, BST] + BFS
17	20240614	[H2424]미생물2	DS : 2.2%	PQ + SqrtDeomp[Seg, BIT]
18	20240622	[H2425]렌터카 조회	DS : 12.3%	Hash + PQ(set)
19	20240712	[H2428]직선열차(py)	DS : 9.7%	BFS(SP)+idea
20	20240720	[]연락처DB2	DS : 4.5%	Hash



# 2025년 pro기출 목록

: 25년 06월 23일 현재 30문제

num	date	SWEA title	result	Algorithm, Data-Structure : C++
1	20250111	[H2502]경유지운송(py)	DS : 15.5%	BFS(dfs)+biSearch or MST(UnionFind:Kruscal, PQ(set):Prim)
2	20250118	[H2503]파일 다운로드2	DS : 4.6%	implement + (Hash, PQ) : 데이터 약함(sample code 반례있음)
3	20250124	[H2504]사각형그리기	DS : 6.6%	Grouping(SqrtDecomp) + UnionFind
4	20250214	[H2506]온라인마트	DS : 2.3%	set(pq)
5	20250222	[H2507]전기차충전소	DS : 0.9%	pq(set) + hash
6	20250228	[H2508]도로연결	DS : 7.0%	Grouping(SqrtDecomp) + UnionFind
7	20250308	[H2510]인기 검색어	DS : 6.7%	Hash + UnionFind, Trie + Dfs(모범코드)
8	20250315	[H2511]음악방송	DS : 1.1%	Grouping(SqrtDecomp) + subSum
9	20250321	[H2512]공항 주차장	DS : 11.1%	PQ(set) + Time_Based+Idea
10	20250405	[H2514]결정 나무	DS : 6.2%	DFS/BFS, (renum_hash)
11	20250411	[H2515]통행료 인상	DS : 6.2%	Dijkstra, idea: Ad_hoc
12	20250418	[H2516]마라톤 코스	DS : 12.8%	DFS/BFS (Path Counting), (renum_hash)
13	20250510	[H2518]광물운송	DS : 12.2%	sqrtDecomp + UnionFind
14	20250523	[H2519]카페2	DS : 19.6%	renumHash + queue + set(pq)
15	20250530	[H2520]지렁이 게임	DS : 4.4%	renumHash + implement
16	20250614	[H2522]도로 파괴	DS : 8.1%	renumHash + Dijkstra
17	20250620	[H2523]단어 사전	DS : 0%	Trie or SqrtDecomp(tolInt, toStr, vector(lower_bound, insert, erase)) + Quotient and Remainder
18	20250628	[H2524]로그인 대기열2	DS : 0%	list + SqrtDecomp(Fenwick:BIT)
19	20250705	[H2526]기계식 주차장	DS : 2% p(3)	indexHeap+bitCheck
20	20250712	[H2527]에너지 운송	DS : 10% c(7)	PQ + popCount



**Thank you**