Problem Solving Sheet	This google sheet is created by **Dr Mostafa Saad Ibrahim**. Overall **~950** problems for **newcomers** to problem solving.	
	The goal of this sheet is to enhance algorthmic probem solving skills for both programming competitions and interviws preparations	
	Currenet Version **V7.0** Latest Version	
	This is read-access only file. Read line 31 to create your OWN COPY.	
	mostafa.saad.fci@gmail.com Site	
	To understand the sheet, watch these videos	
	English Speakers	
	Video Introducing roadmap (Arabic) - to min 18 ONLY	
	New Video explaining the sheet (Arabic)	
	To know how an online judge work: watch 2 videos Online Judge and Codeforces. In near future, finish all of it	
	Once you are **DONE** with this sheet, utilize this supervision general pool or use it for ider topics-based training	
	For guys with interest only in preparing for interviews NOT competitiotns, use this sheet **Interviews**	
Prerequisites?	Programming skills with prefreed ad-hoc problem solving styles (see udemy course below)	
	Any language, but C++ is the most prefrred.	
My Relevant UDEMY courses	**Coupons**	
	Mastering 4 critical SKILLS using C++ 17 Specifically content till project #3	
	Mastering critical SKILLS in Data Structures using C++	
	Mastering critical SKILLS for Algorithms in C++	
	Mastering critical SKILLS for Coding Interviews	
	Mastering 4 critical SKILLS using Python	
Discord Channel	Read the rules. Interact with others	

What is this Sheet?	- Complete and consistent roadmap for newcomers: What to solve & algorithms to learn in order - In the bottom row, there are different sheet pages such as Faq, Topics, C2 - C1, C2 are (Codeforces Div2 C problems (or similar level from other OJs), but from easy to hard). Same for D1, D2, D3 - Covering most of topics needed up to codeforces Div2-D - Problems of scales 1 - 5.5 / 10 + Few harder ones - Problems increase in difficulty per topic with intermediate easy/medium problems + ad-hoc problems - Speed problems to maintain speed goals - A lot of recorded videos for problems solutions, especially for the entry levels (Arabic) - Several students followed its order and managed to solve by themselves 95% of it (up to his current sheet page) - Where are the problems? In the bottom of the sheet find Tabs: A, B, ...and so on
How to train?	- You can train in one of the following ways: - A) **Blind-Order training style** - Problems are distributed in sheets A, B, C1, C2, D1, D2, D3 - This one is a roadmap. It targets learning the knowledge/skills in a consistent and balanced way - Every sheet page is on average harder than the previous sheet page - This is my **recommended** way, though most camps/training-approaches don't use this style - B) **Topics-Based training style** - See sheet page (Topics). It has the same sheet problems (A to D3) ordered by category and level, around 950 problems - **Ideas Quality column**: P5 (important), P4(very interesting), P3(interesting), P2(good), P1(ok), Empty (normal) - Say your level is 6/10, and solved a problem of level 3 with P5, you will find it a normal one. So notice, it is subjective to your level/background - You can train using Blind-Order, and use Topics page as guide to skip some problems - Many guys/training camps are fan of this topics-based way. - You need to be careful with such style as it may corrupt your training quality, e.g. due to your bias - Advantage: Mastering the algorithm till solving some hard problems in a short time - Disadvantage: Discovering the algorithm behind the problem is an important skill. Given that you know the topic, you lose a good space to improve this skill - Disadvantage: Being in the mode of specific algorithm lets you solve many of it easier. However, when solving in real contests, your mind is not so active on the specific topic - It is still a good training roadmap. Actually used by most of people I think.
Advantages of this Sheet?	- To be a strong contestant, one has to take care of a number of quality and quantitiy factors ==> This roadmap does its best to satisfy that - Typical issues in our Arabian region: Guys with 700-1000 solved problems and still weak! - Why? A) No specific roadmap or keep switching between them B) Training while knowing problem category / level C) Focus on specific online judge - Again, this sheet solves these issues - Allows you to write down your statiscs to learn from them (e.g. you consume much time in debugging) - Continuous refining based on feedback

Your Sheet COPY	This is a personal Google sheet for you [**Make a copy** from file MENU] to have sets of problems to solve coupled with algorithms to learn - Don't download the sheet, Use it online - Can't edit it? Because it is **read-only**. Read below notes. - Just make a copy to your google driver - Then work over it online. Following are the details - Login to ur google Gmail - Go to my sheet - In the sheet click on 'file' menu - select Make copy - it will create copy for u - **RENAME** it to Junior Sheet <Your name> - Now the copied sheet is opened for you (or go inside ur Google drive and you will find it) NOTE: If u did so and still read-only format, then you are again opening my sheet (e.g. with old name), NOT your copy
For Whom?	A junior is anyone who does't master solving codeforces Div2-D.
Skills Goals	Moving from Junior Level to Semi-Senior Level: A one who do pretty well in CF-Div2 A, B, C, D and similar levels (e.g. TC-Div2-1000)
Knowledge Goals	Understand and build fair knowledge in some algorithms in Number Theory, Dynamic Programming, Greedy, Graph Theory and Search
Sheets	Sheet pages are mainly for Codeforces Div2 A, B, C, D + Problems on knolwdge topics (Mainly from UVA, SPOJ)
	Each sheet has some sets, each set is ~10-15 problems....The top sets are mandatory....The below sets (after line mark) are optional
	If you did well in the mandatory sets, move to next sheet...otherwise you still need training on similar level...then solve the optional problems
	Please watch the videos in order, solve UVA/SPOJ problems in order. Don't skip them.
	In some columns, some time recordings. This helps you to know how much time you take per a problem...use that to recognize your problems
	In the level column give an estimate to the problem level from 1-2 (easy), 3-4 (medium), 5-6(hard), 7-8 (had to read editorials), 9-10 (can't solve)
	In the comments column..write comments for hard problems.
	Put problem Status | AC (for Accepted) | CS (can't solve) | Other values: WA (wrong answer), TLE (time limit exceeded), RTE, MLE
	If you solved a problem before, put ACX instead of AC. Don't resolve
	Don't let a problem consumes more than 2-3 hours. If can't solve it, see editorials/solutions. If still can't solve it, just leave it for now.
	Don't compare yourself with others. People vary in their progress

Problem Level Column	Use the following guide to assign a proper problem level Div2-A => 1 - 2 Div2-B => 1.5 - 3 Div2-C => 3 - 5.5 Div2-D => 5 - 6.5 Div2-E => 6 - 7.5 Div1-D => 7 - 8.5 Div1-E => 8 - 9.75 In other words, most of the time, one shouldn't assign Div2-A problem level such as 5. But it can be: 1, 1.5, 2. Very few might be 2.5	
Notations	CF136-D2-A	CF (codeforces), D2 (Division 2), (136, A) is the problem URL. Note this is not Round 136 ... it is Round 97
	SRM150-D2-1000	SRM 150 (Topcoder), D2 (Division 2), 1000 (3rd problem)
	For Topcoder: If using the applet arena, then we normally use SRM number (the old way) However, if using the web arena, you need first the problem name itself! (the new way - https://arena.topcoder.com/) The fastest way to get problem name is from the editorial if exist https://apps.topcoder.com/wiki/display/tc/Algorithm+Problem+Set+Analysis [LOGIN here first] https://www.topcoder.com/tc?d1=match_editorials&d2=archive&module=Static https://www.topcoder.com/blog/tag/srm-editorials/ [for recent SRMs] How to see contest submissions: https://www.quora.com/Where-can-I-find-the-solutions-to-the-problems-in-TopCoder Otherwise from match archive: https://www.topcoder.com/tc?module=MatchList For notes and tricks for using Java arena applet: See https://goo.gl/Q43tRL. We can download code with all cases on local machine and code normally. Topcoder nowadays maybe slow. So wait for 5 min for a problem to open. If did not work, try in another day https://codeforces.com/blog/entry/61252	
Problems Colors	CF483-D2-A	White for a problem from codeforces
	UVA 10242	Basic (if possible) Knowledge problem on the just watched videos
	SPOJ CDOWN	A knowledge problem on topic you watched before, will be harder than basic problems
	CF518-D2-B	Problem of easier level than current sheet page level to enhance multiple training levels in same time instead of 1 level training
Moving faster	Do I have to solve every problem? For Div2 (A, B, C1) => No. If you can move faster, do it. For Non CF problems (E.g. UVA), please solve all	

Others Solutions	If you solved a problem, please see some other accepted solutions in codeforces. You don't need to watch my linked videos unless can't solve	
External Resources	Awesome Competitive Programming	Many awesome links - very helpful for English guys
	Ahmed Elsaghir Trainnig	Ahmed is senior from GUC
	A2oj Ladders	Don't like my sheet? Go with Ahmed Aly Ladders
	Prgramming Ahmed M sayd	Arabic Programming Playlist
	Programming Mohamed desouky	Arabic Programming Playlist
	More Resources	Each video is part of a playlist
History	V1: initial release	
	V2: Vidoes updates. Sheet P2A: Little problems replaced + reordering. P2B, P2C, P2D merged in P2B. P3A and P3B: new knowledge sheets	
	V3: Added problem names. P3A, P3B split over 3 sheets, reordered to be more incremental rather than random	
	V4: (https://docs.google.com/spreadsheets/d/12YI86X40xGtid9t1dUHkK6urrqh6nTaPEvOKBkAbAgU/edit?usp=sharing) Solving many knowledge concerns: 1- Discarded rare topics (and their problems): ~20 videos. 2- Adding Easy problems after each video. You don't have to search by yourself anymore 3- Distributing many of the knowledge problems inside the the main sheets instead of delaying them to the last sheets. Other concerns: 4- More smooth transitions from a sheet to another 5- Utilizing the new many problems added by CF since initial sheet creation	
	V5	
	- Added Video Solutions to some existing problems - Added Easier DP problems after its Intro videos - Added new topics: Tree Diameter, Isomorphism, DP (bitmasks, games, probability), Max Flow, SCC, Segment Tree, 2 pointers, Trie, KMP, Geometry Polygons - Added problems for old categories to balance the available problem levels per category. - Added 3 sheets for Div2-D (contains the old Misc sheet problems) Note: If you were using version 4.X, then the major change for you is replacing "Misc" sheet with the 3 Div2-D sheets. If wanna migrate: - Then Remove Misc sheet - Click on the arrow for Div2-D sheets, and make copy for your sheet	

	V6: - Added 3 columns to the sheet: debug time, category and by yourself columns - Each sheet is enhanced with problems from the lower sheet (shifted from it). The purpose is to mix levels per sheet, hence allow multiple training levels in same time (hard vs speed concern). See the new added color - Added probability/expectations English videos/problems - Added Topological sort problems - Solution editorials linked to many non-CF problems / more videos in DIv2-A/Div2-B - Add many problems where my trainees marked as interesting problems. Removed some problems that I think not that interesting or its ideas covered by other problems (subjective). I am working on sheets with a simple, but hard to do idea: Most of the problems seems for the trainer novel in idea with less repeated ideas, hence learning a lot while solving much less. - Add topics based training style sheet page V7: - Added Topics2 (See notes there) - not intended for juniors

Thanks for all guys who sent sheet feedback: Mariam Alshereef, Magdy Hassan, Ahmed Yasser, Ahmed Elsayed Awad, Mohamed Nasser, Mostafa Ali Mansour, Aya elymany, Ayyad shenouda, Others.

Special Thanks for Coach Alhussain Aly for his continuous help

Special Thanks for All volunteers in videos recording / editorials writing