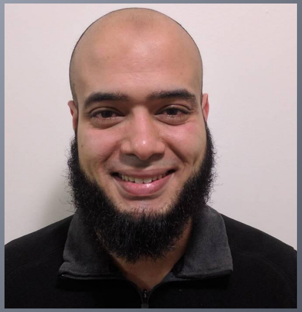
P THINK FAST S

Competitive Programming

From Problem 2 Solution in O(1)

ICPC Community
Building and Training

Mostafa Saad Ibrahim
PhD Student @ Simon Fraser University



Successful Communities

Individual efforts

- Trainees must train hard individually
  - Collaborative training will make it much fun
- Take care of many quality and quantity factors
- Learn from mistakes - make use of others experience

Collaborative efforts

- Do knowledge transfer / mentoring / coaching
- Organize contests / events / camps
- Teach Programming, Data Structures and Algorithms
- Attracting new promising candidates

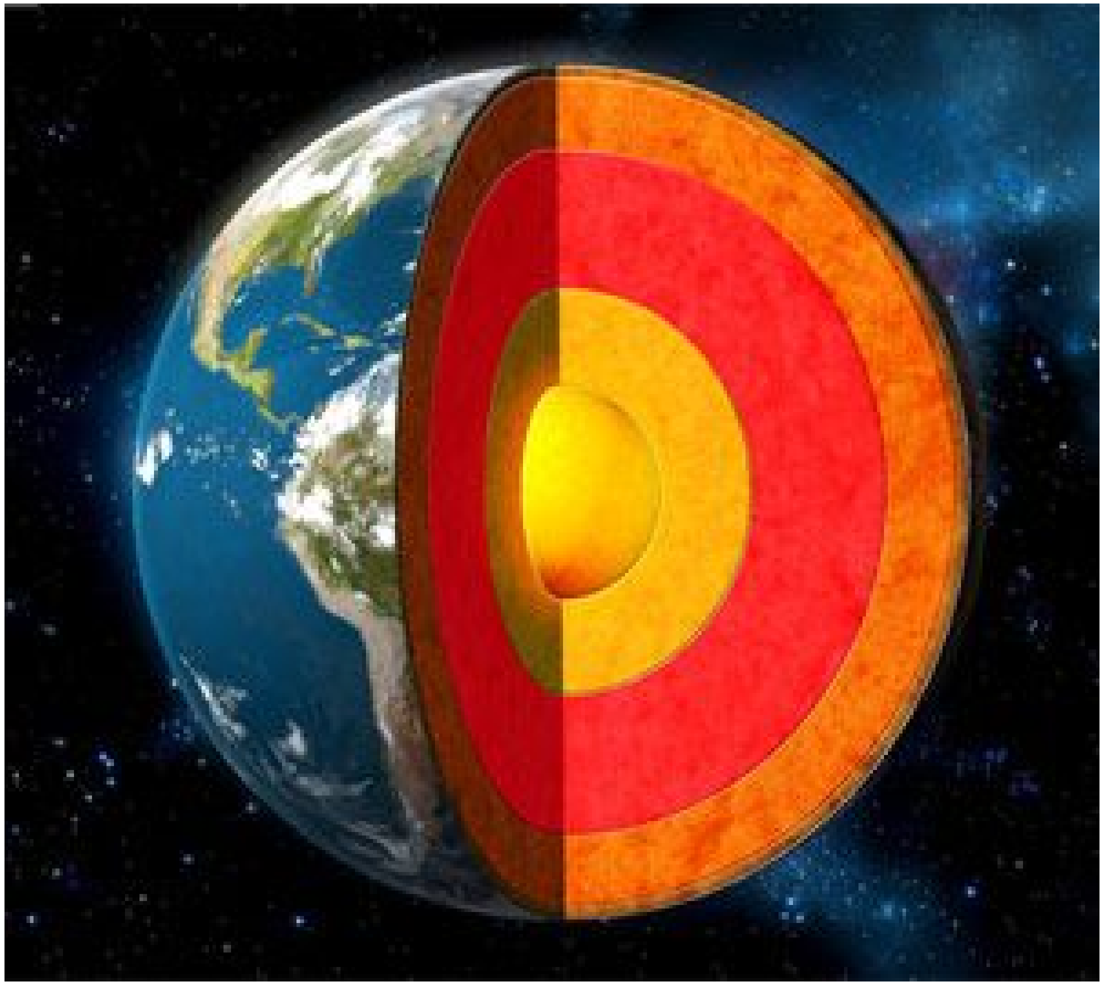
Institute support

- Fund for the events
- Fund for official contests: registration/transportations
- Resources: Permanent / temporary (training / events)
  - Ask for small **permanent lab** for ACM
- Handle date conflicts with exams/assignments

Training common issues

- Individual
  - Guys with 700-1000 solved problems and still weak!
    - No specific roadmap or keep switching between them
    - Training while knowing problem category / level
    - Focus on specific online judge
  - Psychological Issues
- Young Communities
  - Do a lot of ad-hoc training roadmaps / switches
    - Tip: Select one roadmap and keep going with it
  - Few resources / little official support
- *Time to get rid of that!*

Community Building: The **CORE**



Img src

Building The **CORE: Trainees**

- The core (nucleus) is first fundamental part
  - Don’t target getting many people in the first year
  - Just ~6 guys of: **Passion & Hard working**
    - Assume you are the first 1-2 persons
    - Identify guys who are good in programming in your class or other classes (or at least like learning)
    - Convince them why this track is so important
    - Keep going till find ~6 guys of interest/dedication
  - Train together for a complete year : **Follow my** **sheet**
    - My sheet will boost your level within a year
    - After that, you know much about ICPC and Training
    - Your online/onsite performance is key for next years

Building The **CORE: Official Support**

- Find a **<u>doctor</u>** who is so **keen** to help students
    - Explain to him why this training is important
    - What kind of support you may need
    - 2-3 meetings yearly to inform about progress
- Talk to other staff
    - Sometime TAs are barrier and talk negatively
    - If know who do so, convince him why it is important
- More connections
    - Let the core trainees visit the dean, inform him about your efforts and what support they may gain.
    - Better Ask the Dr to come.

Attracting newcomers

- Active Facebook page / contacts
- Teach programming & datastructure courses
  - Identify active students / talk to them
- Events in summer/winter vacations
  - Why problem-solving? First 2 videos here
  - Ask popular figures to present such sessions
    - Many from other communities would like to help
    - Determine specific scope for the talk with speaker
  - Basic thinking questions to stimulate them
    - Binary Search: How to find a page in book?
    - Game theory: Ad-hocs - Nim

Attracting newcomers: Marketizing

- Did some achievement in ECPC/ACPC?
  - Ask Dean to put this news on faculty site
  - Ask Drs of different classes to announce that
  - Announce on faculty walls
  - Party after ACPC to thank teams & draw dreams
- Planning event/contest/training
  - Announce and encourage to attend / prizes
  - Booth in entrance: Stop students & talk to them
  - Community **stars:** talks / Success stories / Camps
- Use Institute Official support as possible
  - *Students trust <u>Doctors/Key figures</u> more than students*

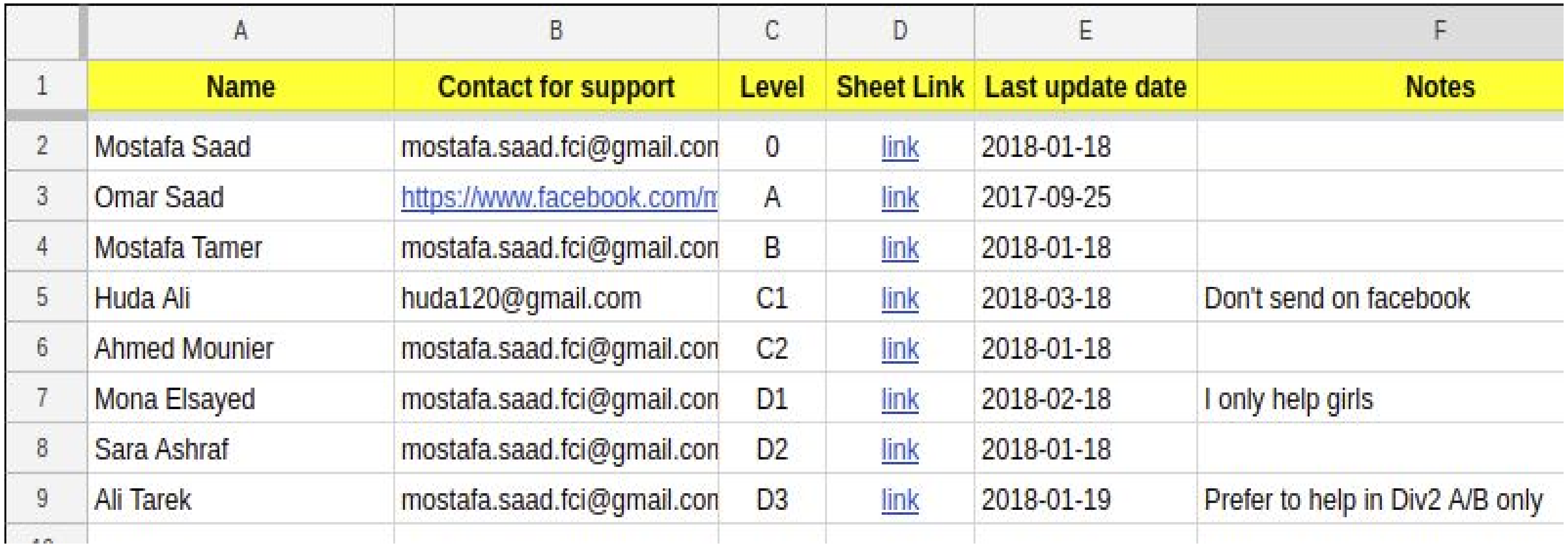
Vision based on my Juniors sheet

- 4 junior levels: Codeforces D2: A, B, C, D
  - Complete roadmap: What to solve & learn + the order
  - 800 problems of scales 1 - 5.5 / 10
  - Covering all topics needed in codeforces D2, **in order**
    - Except few of Div2-E
  - Problems increase in difficulty
  - A lot of recorded videos for problems solutions
  - Several students followed its order and managed to solve by themselves 95% of it (up to his current sheet page)
  - Continuous refining based on feedback
  - By its end, you are a fresh semi-senior
    - *Don't call somene before that stage a semi-senior*

Training Plan

- Each student has his own online sheet copy
  - No skipping for problems - time per [problem]
  - Trainee level: Junior-A, Junior-B, Junior-C, Junior-D
    - Note: sheets are A, B, C1, C2, D1, D2, D3
  - Junior-A prerequisites
    - Programming 1 / STL (or actually much less)
  - Encourage each level to **train together**
    - Have lab. Meet together in the lab in specific time
    - You all solve in your sheets and encourage others
    - **Collaborative training** helps so much
  - Ask for help from one solved the same problem

Training Progress

- Create a sheet to track all trainees (see)
  - Each time a trainee move to new level, update sheet
  - Need help in problem? Asked one who solved it/monitor
- 

	A	B	C	D	E	F
1	**Name**	**Contact for support**	**Level**	**Sheet Link**	**Last update date**	**Notes**
2	Mostafa Saad	mostafa.saad.fci@gmail.con	0	link	2018-01-18	
3	Omar Saad	https://www.facebook.com/n	A	link	2017-09-25	
4	Mostafa Tamer	mostafa.saad.fci@gmail.con	B	link	2018-01-18	
5	Huda Ali	huda120@gmail.com	C1	link	2018-03-18	Don't send on facebook
6	Ahmed Mounier	mostafa.saad.fci@gmail.con	C2	link	2018-01-18	
7	Mona Elsayed	mostafa.saad.fci@gmail.con	D1	link	2018-02-18	I only help girls
8	Sara Ashraf	mostafa.saad.fci@gmail.con	D2	link	2018-01-18	
9	Ali Tarek	mostafa.saad.fci@gmail.con	D3	link	2018-01-19	Prefer to help in Div2 A/B only

Training Sessions: Levels A-D

- Focus n Juniors A & B
  - Sessions topics follow the videos order
  - Encourage them to watch the video 2-3 times
    - If they understood, let them proceed with solving
  - In session, check if they really watched? Stress on that
  - Explain the topic. Solve few problems below video
- For C/D levels
  - I think: Encourage them to depend on themselves
  - Special Sessions to help understanding specific parts
  - Help in hard problems

Training Sessions: Level Zero

- For who knows no/little programming
- Phase 1
  - Variables, IO, conditions, basic looping
  - Videos: My playlist (1-10) or Bucky (1-8 + 16-22)
  - Practice from Assiut CF Problems (=URI here)
- Phase 2
  - More Loops, arrays, and functions
  - Videos: Mine (11-16), Find in bucky - URI Practice
- In parallel: Finish my playlist - sheet Div2-A
  - My sheet will guide to all backgrund videos they need

Training Sessions: Level Zero

- Assiut created problems customized to students background to attract them more
  - Use polygons to create new problems
- *Soon I will add **detailed** sheet page for them*

Monitors

- Split mentors based on the resources
  - e.g., per level - or group of say 5-10 students
  - Encourage students to record the sheet statistics
    - Check their code and give comments
    - Push them to think more (think column)
    - Coding/Debugging column should be 10-15 min
    - Target 1st submission - don't say it is offline solving
  - Monitors should provide good support, but in smart way
  - Don't do everything for the trainee needing help
  - Give some guidelines, then push them to get things done
  - Encourage them to try harder and enjoy the progress
  - Teach them to mark hard things as 'TODOs' / keep going

Contests

- Individual
  - Online contests are important for your behaviour
  - Encourage students to attend 1+ online contest per week
  - Codeforces, AtCoder, CSAcademy, Topcoder, Hackrank..
- Teams
  - Create biweekly contests for them from time to time
    - Use Codeforces, A2oj, Hackrank, ...etc
    - Avoid inventing problems - save your time
  - Encourage them to build teams / how to build them / ..
  - Let them try different people teams configuration
  - Announce winners - little prizes - marketize

ICPC Contests: Be ready

- Teams final configuration 3 months b4 contest
  - Watch contest videos
  - Build strong teams / diverse in topics
  - Run some individual contests
    - Measure statistics: How many from 1st submission
    - Sort people based on stats
    - Form initial teams based on that
    - Advise members on what is good for the teams
    - There many other ways: Ask E.g. in GUC/AinShams
  - Do lots of team contests
    - Each team prepare his single own library
    - Each team practice his own strategy

Community Continuity

- One of the real challenges is how to keep the *community strong*
  - E.g. FCICU was strong for some time, but not nowadays
  - Real Leaders think about the **long-term sustainability**
- Community Structure and Elections
  - Define structure per year or term
  - Elect the people / discuss sustainability, goals, milestones
- Teach new trainees to give back the favor
  - Come and help others - dedicate time
  - Nothing like internal feelings of helping others

Beyond technical concerns

- Psychological Issues
  - Typically trainees suffer from such issues
  - Mentors / Leaders: Keep talking/advicing with them
  - *Frustrations (slow progress, can't solve, online perform)*
  - *Worrying about the Appearance & Failure*
  - *'Should I stop' Dilemma?*
  - *Seniors: 1st or nothing / No breaks*
  - *Psychological barriers: Hating probability/geometry*
  - *Many more!*
- Study-ACM Balance!
  - Give them tips - Read

Messages for juniors

- Messages
  - Secrets of Success
  - You can be great by yourselves
  - You don’t need massive follow up
    - I know guys finished the whole by themselves
  - You need to try seriously before asking for help
  - Don’t be shy to ask others (on FB or whatever)
  - Don’t compare yourself to others
  - Participate in contests - don’t think in your ranking image
  - **Return the favor: Transfer knowledge - help others**

Misc

- Another Path for community building
  - After building the core
  - Go and ask one of the available community to help you **duplicate** their training
    - E.g. tells you what to teach, which problems to use, what levels to divide students E.g. Ain Shams, Assiut

Acknowledgement

- Thanks to helpful suggestions :)
  - Eng Islam Al Aarag (GUC)
  - Eng Sara Elkadi (Mansoura)
  - Eng Ayman Salah / Eng Hussien Ibrahiem (Assiut)

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً