

- Я Um\_nik, и ниже описан мой взгляд. Это мое мнение. И только. Не больше, но и не меньше.
- Я утрирую. Все нужно воспринимать через призму вашего опыта, конкретной ситуации на констесте и специфику вашей команды. Вы можете попробовать применить рекомендации отсюда as is НА ТРЕНИРОВКАХ и посмотреть, что вам подходит.
- Это про командные соревнования по правилам ICPC. Я ничего не говорю о том, как решать задачи. Какие-то вещи применимы и к личным соревнованиям, но я не буду на этом заострять внимание.
- Используйте мои рекомендации на свой страх и риск.

## Базовая философия

- Нужно экономить компьютерное время.
- Нужно обсуждать задачи.
- Не стоит быть слепым оптимистом.

## Компьютерное время

- ВСЁ, что можно сделать не за компом, нужно делать не за компом. ВСЁ!
- Нужно всегда продумывать, что ты собираешься писать, ПЕРЕД тем, как садиться за комп. За компом нужно писать код, поэтому нужно подготовиться к моменту, когда доступ к компу появится. Не нужно опускаться до мелочей, продумайте глобальную структуру, декомпозицию на функции, задумайтесь о том, где могут быть сложные места, о них подумайте поподробнее.
- Если вы заметили, что думаете за компом, - свалите с компа, скажите сокомандникам, что вам надо пару минут подумать.
- Оцените время на написание кода, сообщите его команде. Если вы нормально продумали, что именно вы собираетесь писать, это не должно быть большой проблемой. Поначалу будет непросто, но это полезный навык.
- За компом время летит незаметно.
- Если не следить за этим специально, то 80% времени за компом уходит на моменты, когда вы не пишете код, а думаете о том, какой код нужно написать. И это нужно было сделать заранее! А если не сделали заранее, то свалите с компа.
- Ваш сокомандник за компом не замечает, как летит время; поэтому сказать "да ты охренел, ты уже час пишешь два бфса" должны вы. Поэтому он должен был сказать, сколько времени ему понадобится на написание.
- Дебагаться за компом нельзя! Распечатайте код и свалите с компа.
- Иногда компьютер нужен для каких-то вспомогательных действий. Например, написать какой-то перебор и посмотреть на его результаты, чтобы лучше разобраться в задаче. Это нормально, обычно этот вспомогательный код достаточно простой и короткий, но его все так же нужно продумать до написания.

- После сабмита освободите комп. Сразу, а не после получения вердикта. Вердикт вам сообщат через 5 минут, займитесь пока чем-то полезным. Чем бы вы занялись, если бы получили АС? - вот этим и займитесь.

## Взаимодействие

- Нужно знать свои сильные и слабые стороны, а также сильные и слабые стороны сокомандников. Это можно использовать при раздаче задач. Иногда сложные задачи существуют на стыке двух (или более) областей, очень классно, когда можно сказать "Смотри, я свёл к такой задаче, теперь это вообще не про строки, просто какие-то структуры данных, ты же в этом шарить?"
- Нужно знать, какие задачи нравятся и не нравятся вашим сокомандникам. Всё из предыдущего пункта можно отнести и сюда. "Нравится" - это вообще очень классный критерий задачи. Получать удовольствие от процесса важно само по себе, но это может еще и улучшить результат.
- Не бойтесь менять задачи. Когда уходите с задачи, постарайтесь кратко описать сделанные продвижения (например прямо на условию) или рассказать задачу тому, кто её перенимает.
- Вообще, рассказывайте об интересных идеях сокомандникам! Не стоит отвлекать человека за компом без надобности, но рассказать о значительном продвижении третьему - это хорошо. Он может предложить взгляд со стороны, да и просто процесс рассказывания идеи может помочь лучше в ней разобраться или найти ошибку.
- Достаточно сложные задачи перед написанием нужно рассказывать сокоманднику, иногда вместе с высокоуровневым описанием реализации. Вам все равно надо продумать, что писать, так расскажите это сокоманднику: может, он предложит более простой способ.
- Задачу должен писать не тот, кто её придумал, а тот, чье написание принесет большую пользу. Критерии пользы могут быть разными: может быть человек просто быстрее пишет код, может быть лучше знает необходимый алгоритм, может быть понимает решение лучше, чем автор, а может быть он просто менее полезен в оставшихся задачах. Если вы знаете сильные и слабые стороны сокомандников, вы должны быть способны определить наиболее подходящего человека.
- ГОВОРИТЕ ВСЛУХ | РИСУЙТЕ КАРТИНКИ

## Дебаг

- Будьте готовы к тому, что вы будете ошибаться. Когда вы дописали код, в нём есть баги. Не надо отчаиваться, надо их исправлять.
- Дебагаться за компом нельзя! Распечатайте код и свалите с компа.
- Дебаггер - зло. Вы просто сидите за компом и бездумно тыкаете кнопку. Заранее подумайте, на значения каких переменных в какие моменты исполнения программы вы хотите посмотреть, выведите их... Распечатайте и свалите с компа!
- Научитесь читать код.

- Научитесь читать не только свой код, но и код своих сокомандников.
- Научитесь писать код так, чтобы у ваших сокомандников не вытекали глаза.
- Не вносите правки по одной, и не сабмитите после каждой правки. Дочитайте код до конца, сделайте все правки на листочке, потом внесите их сразу все. Не забудьте запустить код на всех сэмплах перед сабмитом.
  - Да, даже если вы точно нашли ТОТ САМЫЙ БАГ, из-за которого всё не работало, всё равно дочитайте до конца.
- После пары прочтений кода (а иногда и до) можно и нужно писать стресс. Стресс-тест - это тестирование вашего решения на большом количестве случайных маленьких тестов. Чтобы это делать, вам нужно будет написать генератор тестов и решение, которое правильно работает для маленьких тестов. Делать это нужно, если есть очень простое решение, которое вы можете написать быстро и уверенно. Все рекомендации из раздела про компьютерное время применимы и здесь.
  - Тут конечно есть конфликт. Вроде как надо в любых ситуациях экономить компьютерное время, значит: читай код вместо этого. Но иногда без стресса очень тяжело, и время, потраченное на стресс, окупается с лихвой. Когда пришло время писать стресс - это творческий вопрос.
- Многие новички, а иногда и опытные, не пишут стресс-тесты, т.к. считают их чем-то сложным. Нужно правда привыкнуть писать их часто. Чтобы понимать, в каких задачах стресс поможет, а в каких нет, нужно проверить это на тренировках.
- Убедитесь, что решение правильное, для этого расскажите его сокоманднику (может быть, в больших деталях, чем раньше).
- Перечитайте условие. Может быть, вы не заметили какую-то особенность вывода, а может быть есть частный случай.
- ЕЩЕ РАЗ перечитайте условие.
- Время на дебаг оценить очень сложно. Можно оценить время на написание и даже на придумывание задачи, но не на дебаг. Не рассчитывайте, что если на написание вам нужно 20 минут, то через 20 минут задача будет сдана. Однако через 20 минут вы освободите комп, и это хорошо.

## Тактика

- Договоритесь о том, как вы принимаете решения. Это может быть демократия, это может быть диктатура, это может быть бросок монетки. Важно, что когда решение принято, ему должны подчиняться все.
- В идеале все должны знать, что сейчас происходит, то есть чем занимаются все члены команды. По крайней мере все, кто принимают решения. Расскажите вашей монетке, что вы собираетесь делать, дайте ей осознанно упасть на нужную сторону.
- Монитор - важный инструмент, он может серьезно помочь с поиском простых задач. Однако не нужно из-за него убиваться. Не нужно втроем решать одну задачу только потому что её сдали 30 команд. Случается всякое. Может быть это боян, а вы его не видели. А может быть просто у вас что-то переклинило. Отвлекитесь, порешайте другие задачи, вернитесь к этой через полчаса.
- Нужно следить за временем. Если вы посмотрели время, скажите его вслух.

- Знайте свои амбиции. "Слабые команды не могут себе позволить читать все задачи". С другой стороны, сильные команды не могут позволить себе роскоши следовать за кем-то.
  - Прочитайте все задачи рано, если вы собираетесь выигрывать. Через час-полтора все члены команды должны знать условия всех несданных задач. После сдачи халяв устройте сессию, когда вы по порядку рассказываете условия задач (тот, кто читал данную задачу) и обмениваетесь первыми впечатлениями (в какую сторону думать, насколько сложной кажется задача).
  - Не стесняйтесь играть по монитору, если вы сильно слабее лидеров.

## Последний час

- "Последний час" (это термин, это могут быть не 60 минут) - это особое время, в которое многие правила утрачивают вес.
- Вы должны принять решение о том, какие задачи вы собираетесь СДАТЬ до конца контеста. Основывайтесь не на идеальном течении, а на реалистичном. Вы можете сделать план на случай "ну а если мы все это быстро сдадим", но его реализация не должна начинаться раньше, чем вы все сдадите.
- Выбросите все задачи, которые вы не собираетесь сдавать. Да, даже если "там немного осталось додумать". И даже если там есть придуманное решение.
- Вы должны максимизировать результат. Результат - это количество задач, сданных ЗА ВРЕМЯ КОНТЕСТА. Лучше вы сдадите одну задачу в 4:59, чем две в 5:01.
- Обычно после написания задачи есть какое-то будущее, о нем нужно думать и им можно пользоваться. В последний час это не так. Из этого есть много следствий, основных два:
  - у вас не будет полчаса на вычитывание потом;
  - ваши сокомандники не должны быть заняты придумыванием задач на будущее.
- Now kiss! Используйте освободившихся сокомандников для вычитывания прямо сейчас. Это называют парным программированием, хотя я чаще употребляю термин "писать код вдвоем". Второй человек сидит за плечом, сразу пытается заметить ошибки, а также с ним можно обсудить что-то, если возникло недопонимание "а что писать дальше?" Это реально работает, не раз я так сдавал 300 строк с компила.
  - Можно и наоборот: лучше понимает решение тот, кто не за компом. А человек за компом умеет "написать Дейкстру", пока другой думает.
- Если вы не стали писать вдвоем (например вы начали писать код до того, как освободился сокомандник), то примерно то же самое, хоть и с меньшим успехом, можно сделать асинхронно. Печатайтесь каждый раз, когда дописали осмысленный кусок кода, пусть сокомандники читают.
- Есть примеры успешных команд, которые вообще все задачи пишут вдвоем, но я бы такое не рекомендовал.
- Если это последняя задача, то после сабмита нужно вести себя так, будто вы получили не AC. Ожидаете WA? - пишите стресс. Ожидаете TL? - сгенерьте

макстест. Уже есть стресс? - увеличьте параметры генерируемых тестов, запустите стресс ещё раз.

## Инструменты

- Сделайте табличку с кратким описанием задач и состоянием относительно каждого участника. Мы отмечаем прочитана ли задача, кто собирается её писать, дописана ли задача. Табличка помогает следить за текущим состоянием, искать непрочитанные задачи, а также быстро вспоминать задачи.
- Не смотрите в standings. Вам нужно получить информацию о том, какие задачи сколько команд сдали. На некоторых платформах (Codeforces) для этого есть специальная вкладка "Основное", на других (AtCoder, Yandex.Contest, ejudge) вверху или внизу standings есть посчитанные числа. Перепишите эти числа в вашу табличку и закройте standings. Выберите на эту роль человека, который наименее подвержен волнению из-за того, что вы слишком хорошо/плохо выступаете.
- Хорошее время для этого - сразу после сабмита. У вас уже открыт браузер, и пока что никто не глубоко в работе за компом.
- При чтении условий можно использовать маркеры для выделения текста и выделять важные места. Во-первых, вы более внимательно ищете важные места, что хорошо. Во-вторых, по раскрашенному условию проще рассказать задачу сокоманднику.
- Нельзя писать на обратной стороне условия! В момент, когда вы перевернули условие нерешенной задачи, она стала неотличима от черновика, вы потеряли задачу и не сдали ее!
- Сделайте так, чтобы сданные задачи на вид отличались от несданных. Сданные задачи можно использовать в качестве черновиков.

## Разное

- Печатать можно не только решения. Если вы умудрились наблуживать в тупняке для стресса - распечатайте его. Если стресс ничего не находит - распечатайте генератор. Печатайте вывод программы, в том числе аутпут дебаг. Печатайте таблички значений, в которых вы хотите поискать закономерность.
- Не используйте на личных контестах макросы, которых нет в командном шаблоне. Если они очень нужны - добавьте в командный шаблон. Если не очень - не используйте на личных.
- Есть красивые истории о том, что сокомандников банили на GCJ, потому что у них настолько похож кодстайл, что они одну из задач написали одинаково символ в символ. Это конечно романтично, но вполне можно понимать код сокомандников даже если он не совпадает с вашим побуквенно.
- Обсуждайте тактику, обсуждайте допущенные ошибки, ищите решения вместе, ходите в фастфуд после конкурса. Разговаривайте, короче!
- Получайте удовольствие!