

Концепция разработки задач

Григорий Шовкопляс

14 июля 2016 г.

С чего начать?

- Придумать задачу (внезапно)

С чего начать?

- Придумать задачу (внезапно)
- Понимать формально, чего вы хотите от участника

Написание условий

При написании условий нужно руководствоваться несколькими свойствами и учитывать их именно в такой очередности:

- 1 Формальная корректность

Написание условий

При написании условий нужно руководствоваться несколькими свойствами и учитывать их именно в такой очередности:

- 1 Формальная корректность
- 2 Понятность

Написание условий

При написании условий нужно руководствоваться несколькими свойствами и учитывать их именно в такой очередности:

- 1 Формальная корректность
- 2 Понятность
- 3 Грамотность

Написание условий

При написании условий нужно руководствоваться несколькими свойствами и учитывать их именно в такой очередности:

- 1 Формальная корректность
- 2 Понятность
- 3 Грамотность
- 4 Смешнявость

Полезный совет №1

Нарисовал картинку — понятность условия возросла в разы.

Полезный совет №2

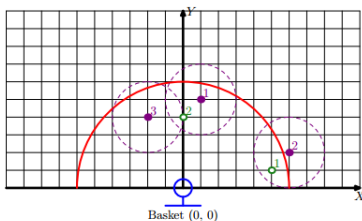
Пояснил примеры — не получил кучу кларов.

Пример из жизни

Пример

basket.in	basket.out
3 2 6 2	2 3 1
1 5	
6 2	
-2 4	
5 1 2	
0 4 3	

Комментарий



На картинке изображен пример. $b_1 = \frac{2}{26 \cdot 4} = \frac{1}{52}$, $b_2 = \frac{3}{40 \cdot 3} = \frac{1}{40}$, $b_3 = \frac{2}{20 \cdot 4} = \frac{1}{40}$.
Обратите внимание, что кольцо является точкой.

Что дальше?

- Пишем авторское решение

Что дальше?

- Пишем авторское решение
 - Не нужно его вылизывать и оптимизировать

Что дальше?

- Пишем авторское решение
 - Не нужно его вылизывать и оптимизировать
- Пишем дополнительные решения

Что дальше?

- Пишем авторское решение
 - Не нужно его вылизывать и оптимизировать
- Пишем дополнительные решения
 - Брутфорс решение.

Что дальше?

- Пишем авторское решение
 - Не нужно его вылизывать и оптимизировать
- Пишем дополнительные решения
 - Брутфорс решение.
 - Решение за лучшую асимптотику, которое должно валиться.

Что дальше?

- Пишем авторское решение
 - Не нужно его вылизывать и оптимизировать
- Пишем дополнительные решения
 - Брутфорс решение.
 - Решение за лучшую асимптотику, которое должно валиться.
 - Вот тут надо бы использовать все неасимптотические оптимизации, что придумаете.

- Существует другое решение за ту же асимптотику

- Существует другое решение за ту же асимптотику
 - Напиши!

Еще решения

- Существует другое решение за ту же асимптотику
 - Напиши!
- Видишь, где можно явно набагать?

- Существует другое решение за ту же асимптотику
 - Напиши!
- Видишь, где можно явно набагать?
 - Напиши!

- Существует другое решение за ту же асимптотику
 - Напиши!
- Видишь, где можно явно набагать?
 - Напиши!
- У тебя есть кореш и ему нечем заняться?

- Существует другое решение за ту же асимптотику
 - Напиши!
- Видишь, где можно явно набагать?
 - Напиши!
- У тебя есть кореш и ему нечем заняться?
 - Пусть напишет доп!

- Тесты должны валить неправильные решения и не валить правильные (внезапно).

- Тесты должны валить неправильные решения и не валить правильные (внезапно).
- Первые несколько тестов должны совпадать с тестами из условия.

- Тесты должны валить неправильные решения и не валить правильные (внезапно).
- Первые несколько тестов должны совпадать с тестами из условия.
- Должен быть написан генератор.
 - Программа, которая создает тесты.
 - Обязательно должна быть возможность поднять ограничения и регенерировать тесты.
 - Надо бы сделать его кроссплатформенным (в разных системах, переводы строк по-разному).

- Тесты должны валить неправильные решения и не валить правильные (внезапно).
- Первые несколько тестов должны совпадать с тестами из условия.
- Должен быть написан генератор.
 - Программа, которая создает тесты.
 - Обязательно должна быть возможность поднять ограничения и регенерировать тесты.
 - Надо бы сделать его кроссплатформенным (в разных системах, переводы строк по-разному).
- Тесты должны быть осмысленные.

- Тесты должны валить неправильные решения и не валить правильные (внезапно).
- Первые несколько тестов должны совпадать с тестами из условия.
- Должен быть написан генератор.
 - Программа, которая создает тесты.
 - Обязательно должна быть возможность поднять ограничения и регенерить тесты.
 - Надо бы сделать его кроссплатформенным (в разных системах, переводы строк по-разному).
- Тесты должны быть осмысленные.
- Не очень круто генерировать только макстесты и ставить их сразу за сэмплами.

Пример

Есть задача про дерево. В тестах должны быть деревья:

- Бамбук

Пример

Есть задача про дерево. В тестах должны быть деревья:

- Бамбук
- Расческа

Пример

Есть задача про дерево. В тестах должны быть деревья:

- Бамбук
- Расческа
- Звезда

Пример

Есть задача про дерево. В тестах должны быть деревья:

- Бамбук
- Расческа
- Звезда
- Метла

Пример

Есть задача про дерево. В тестах должны быть деревья:

- Бамбук
- Расческа
- Звезда
- Метла
 - Сделать всегда ручку длины $n/2$ некруто.
 - Желательна разумная параметризация: ручка длины l , всего ребер m .

Пример

Есть задача про дерево. В тестах должны быть деревья:

- Бамбук
- Расческа
- Звезда
- Метла
 - Сделать всегда ручку длины $n/2$ некруто.
 - Желательна разумная параметризация: ручка длины l , всего ребер m .
- Рандомный Прим

Пример

Есть задача про дерево. В тестах должны быть деревья:

- Бамбук
- Расческа
- Звезда
- Метла
 - Сделать всегда ручку длины $n/2$ некруто.
 - Желательна разумная параметризация: ручка длины l , всего ребер m .
- Рандомный Прим
- Рандомный Крускал

Установка ограничений на TL

- Ниже чем 2x от авторского незаконно, за такое палками бьют.

Установка ограничений на TL

- Ниже чем 2x от авторского незаконно, за такое палками бьют.
- 5x шикарно и универсально.

Установка ограничений на TL

- Ниже чем 2x от авторского незаконно, за такое палками бьют.
- 5x шикарно и универсально.
- Можно поставить среднее геометрическое между авторским и вылизанным неоптимальным.

Установка ограничений на TL

- Ниже чем 2x от авторского незаконно, за такое палками бьют.
- 5x шикарно и универсально.
- Можно поставить среднее геометрическое между авторским и вылизанным неоптимальным.
- Но тут на самом деле все довольно индивидуально и часто ставят тупо две секунды.

Клевые советы

Пссс... Парень, отсек не получается?

Увеличь ограничения!

Java

Просто помни о ней.

Полезная так-то штука

- Генератор должен быть максимально рандомный.
- Решение максимально простое и наивное, как маленькая девочка.

- Должен проверять все, что обещает условие.

- Должен проверять все, что обещает условие.
 - Прямо вот совсем все!

- Должен проверять все, что обещает условие.
 - Прямо вот совсем все!
 - Да, даже решение нужно внутрь встроить, если это необходимо.

- Должен проверять все, что обещает условие.
 - Прямо вот совсем все!
 - Да, даже решение нужно внутрь встроить, если это необходимо.
 - Исключение: то, что нельзя проверить за разумное время (существование гамильтонова пути)

- Должен проверять все, что обещает условие.
 - Прямо вот совсем все!
 - Да, даже решение нужно внутрь встроить, если это необходимо.
 - Исключение: то, что нельзя проверить за разумное время (существование гамильтонова пути)
- Должен, в случае чего, сообщать понятные вердикты.

Кулстори от Глебани

Как-то раз в ЛКШ стали переносить задачи в Polygon. Написали валидатор, обнаружили, что в задаче windows (ДО + scanline) ограничения больше в два раза, чем в условии. Ой!

- Должен проверять ответ участника на корректность и сравнивать с ответом жюри.

- Должен проверять ответ участника на корректность и сравнивать с ответом жюри.
- Не должен падать. Например, если подали отрицательный индекс.

- Должен проверять ответ участника на корректность и сравнивать с ответом жюри.
- Не должен падать. Например, если подали отрицательный индекс.
- Авторский ответ тоже осмысленно проверить на корректность.

- Должен проверять ответ участника на корректность и сравнивать с ответом жюри.
- Не должен падать. Например, если подали отрицательный индекс.
- Авторский ответ тоже осмысленно проверить на корректность.
- Если у участника ответ оптимальнее:

- Должен проверять ответ участника на корректность и сравнивать с ответом жюри.
- Не должен падать. Например, если подали отрицательный индекс.
- Авторский ответ тоже осмысленно проверить на корректность.
- Если у участника ответ оптимальнее:
 - Ни в коем случае не WA.
 - Кидаем FAIL.

- Должен проверять ответ участника на корректность и сравнивать с ответом жюри.
- Не должен падать. Например, если подали отрицательный индекс.
- Авторский ответ тоже осмысленно проверить на корректность.
- Если у участника ответ оптимальнее:
 - Ни в коем случае не WA.
 - Кидаем FAIL.
- Вердикты должны быть осмысленными.

Кулстори с работы Гриши

Чекер был примерно такой: Если «YES», то сделай то, если «NO», то другое. Таким образом `print("Easy")` на изичах «проходил» все тесты.

Кулстори от Глебани[2]

«Чувак проверяет сертификат на корректность, проверяет ответ и доволен.» Но логично, что надо бы еще проверить ответ на соответствие сертификату.

Вопросы?