

Проверка программ в системе SoftGrader

Проверка программ будет производиться дистанционно, в автоматизированном режиме, с использованием системы SoftGrader (<http://softgrader.itmm.unn.ru>). До установленного срока сдачи студентам будет предоставляться доступ для проверки успешности сборки, корректности и производительности разработанных программ. Термин «проверка» означает установление факта, что *готовая работающая программа* (разработанная и отлаженная заранее) собирается и корректно работает в тестирующей системе. Система SoftGrader *не предназначена* для разработки и отладки.

Количество запусков программы в системе не ограничено. Но не стоит этим злоупотреблять, т.к. некоторые тесты занимают продолжительное время и при массовом запуске может образоваться очередь задач. Результаты промежуточных запусков видны только вам. Для того, чтобы продемонстрировать результаты запуска финальной версии вашей программы преподавателю, вам нужно выделить соответствующее решение в списке и нажать кнопку "Select for grade"; демонстрируемое решение будет выделено цветом.

Для финальных версий всех программ будет выполняться просмотр кода. При обнаружении явных случаев подгонки под ответ или манипуляций со временем работы, а также в случае обнаружения заимствования кода других студентов задание будет считаться несданным. В случае возникновения спорных ситуаций / обнаружения ошибок в тестирующей системе для соответствующих задач может быть применена ручная проверка. Все требования к оформлению программ и способ проверки корректности сохраняются.

Методы должны быть реализованы в общем виде, как описано в задании. Проверка корректности с помощью тестовых задач состоит в вызове метода для набора тестовых данных; не допускается жесткое внесение параметров непосредственно в реализацию метода. Входные данные для проверки корректности и производительности подобраны так, чтобы время проведения одного теста было порядка 10 секунд. Указанные в заданиях граничные значения размера задач и численные границы для проверки корректности стоит рассматривать как ориентировочные. Они могут быть индивидуально изменены в случае, если разумная реализация будет работать слишком быстро или долго, или не будет удовлетворять слишком жестким требованиям по корректности.

Эффективность параллельной программы будет проверяться по отношению к ее однопоточной версии. Данная проверка будет проводиться на входных данных, для которых время работы однопоточной версии превышает 0.1 секунды. Эффективность масштабирования при выполнении на N ядрах определяется как $E(N) = T(1) / (N * T(N))$, где $T(1)$ – время однопоточной версии, $T(N)$ – время параллельной.

Обратите внимание, что ограничение по времени, указанное в постановке задачи, относится к последовательной программе. Для параллельной программы ограничение по времени будет уменьшаться в соответствии с числом задействованных потоков.

Для доступа к тестирующей системе SoftGrader вам нужно:

- зарегистрироваться на сайте <http://softgrader.itmm.unn.ru> (при регистрации вам нужно будет выбрать номер вашей группы). Свою Фамилию и Имя указывайте на русском языке!
- войти в систему с полученным логином/паролем
- выбрать «Университетские курсы»
- зарегистрироваться на курс «Параллельные численные методы»

Для проверки реализации вашего метода нужно:

- Выбрать задачу, соответствующую своему варианту (например, «Разложение Холецкого»)
- Ознакомиться с требованиями к формату заголовка функции, решающей данную задачу
- Загрузить на сервер исходный код вашей функции, написанной в соответствии с требованиями (либо в виде файла, либо в виде текста программы)
- Посмотреть на результаты тестирования на вкладке «Результаты»
- Выделить результат, который вы хотите продемонстрировать преподавателю, и нажать кнопку **Select for grade**.

Срок выполнения заданий – 3-4 недели с момента рассмотрения соответствующей темы на лекциях! Данные сроки примерно соответствуют сдаче первой работы в конце марта, второй работы – в конце апреля, третьей работы – в конце мая.

После успешной сдачи программы в системе SoftGrader вами должен быть подготовлен отчет, в котором (кроме стандартных разделов «Введение, Заключение, Список литературы») по каждой из решенных задач рекомендуется представить следующее:

- постановка задачи
- метод решения задачи
- теоретические свойства последовательного алгоритма
- описание реализации последовательного алгоритма (в том числе описание используемых структур данных)
- различные способы распараллеливания, их сравнение
- реализация параллельного алгоритма с использованием общей памяти
- описание тестовых задач, на которых проверялась корректность работы алгоритма
- результаты экспериментов

Отчет достаточно представить в электронном виде (прислать по почте) в течение 1-2 недель после сдачи программы. Для примера прикладываю оглавление одного из отчетов.

Дополнительная информация по работе с системой SoftGrader

Алгоритм решения задачи из вашего варианта следует оформить в виде отдельного C/C++ файла, в котором:

- должна быть функция с фиксированным заголовком, решающая поставленную задачу;
- можно подключать стандартные заголовочные файлы;
- можно использовать дополнительные функции, классы, структуры, шаблоны, lambda-функции, исключения, STL;
- можно использовать и все остальное, что разрешено в C++ в рамках одного файла;
- не рекомендуется использовать ассемблерные вставки;
- не допускается использование вывода на консоль или в файл.

Размер исходного файла должен составлять не более 64 килобайт.

При запуске программы следует выбирать компилятор MS VC OpenMP (поддержка компилятора g++ OpenMP – в стадии разработки).

В процессе работы система может выдавать следующие сообщения в столбце "Вердикт" на вкладке "Мои решения":

- В очереди (Waiting queue) // Программа находится в очереди
- Успешная компиляция (Compilation success) // Программа скомпилировалась без ошибок
- Ошибка компиляции (Compilation error) // Программа не скомпилировалась, так как возникли ошибки
- Выполняется на тесте [id_test] (Running on test [id_test]) // В данный момент решение запущено на тесте id_test
- Успех =((Accepted) // Решение прошло все тесты на корректность
- Тест [id_test] засчитан (Test [id_test] accepted) // Тесты с 1-ого по id_test проверены и программа на них отработала корректно.
- Неправильный ответ (Wrong answer) // Ваша программа вернула неправильный ответ на некотором тесте
- Некорректный вывод (Presentation error) // Ваша программа некорректно выводит данные
- Превышение ограничения памяти (Memory limit exceed) // Ваша программа запросила памяти больше, чем разрешено в ограничении задачи
- Превышение ограничения времени исполнения (Time limit exceed) // Ваша программа выполнялась дольше ограничения по времени в данной задаче
- Ошибка исполнения (Runtime error) // Ваша программа завершилась с ненулевым кодом возврата
- Ошибка простоя (Idle limit exceed) // Если ваша программа не тратит процессорного времени в течение 5 секунд
- Всё плохо =((Deadly error) // Ошибка в тестирующей системе. Прошу сообщать вас о таких ошибках.

В верхней части страницы с результатами работы вашей программы есть две диаграммы, на которых приводится сводная статистика по тестам на корректность и эффективность.

Для корректной программы на диаграмме "Вердикты" должно быть 100% TA (True Answer). В случае ошибки ряд тестов будет отображен с вердиктом WA (Wrong Answer).

Для эффективной параллельной программы на диаграмме "Эффективность" должно быть 100% EF (Effective). Если ваша параллельная программа написана не оптимально, то ряд тестов будет отображен с вердиктом NE (Not Effective).