МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
БОРИСОГЛЕБСКИЙ ФИЛИАЛ
(БФ ФГБОУ ВО «ВГУ»)

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
Решение олимпиадных задач

1. Код и наименование направления подготовки:

44.03.01 Педагогическое образование

2. Профиль подготовки:

Информатика и информационные технологии в образовании

3. Квалификация (степень) выпускника:

Бакалавр

4. Форма обучения:

Заочная

5. Кафедра, отвечающая за реализацию дисциплины:

Кафедра прикладной математики, информатики, физики и методики их преподавания

6. Составитель:

В. В. Волков, кандидат физико-математических наук, доцент

7. Методические указания для обучающихся по освоению дисциплины

Приступая к изучению учебной дисциплины, прежде всего обучающиеся должны ознакомиться с учебной программой дисциплины. Электронный вариант рабочей программы размещён на сайте БФ ВГУ.

Знание основных положений, отраженных в рабочей программе дисциплины, поможет обучающимся ориентироваться в изучаемом курсе, осознавать место и роль изучаемой дисциплины в подготовке будущего педагога, строить свою работу в соответствии с требованиями, заложенными в программе.

Основными формами контактной работы по дисциплине являются практические и лабораторные занятия, посещение которых обязательно для всех студентов (кроме студентов, обучающихся по индивидуальному плану).

На практически занятиях рекомендуется активно участвовать в анализе решаемых задач, обсуждении алгоритма их решения, выборе способов реализации алгоритма на языке программирования. При возникновении затруднений в решении задач важно сразу выяснить все непонятные моменты, задав вопрос преподавателю.

В ходе выполнения лабораторных работ рекомендуется пользоваться рекомендованной в рабочей программе дисциплины литературой и записями с практических занятий. При необходимости, за справочной информацией по языку программирования рекомендуется обращаться к встроенной справке среды разработки или к онлайн-справочникам. Важно при решении задач придерживаться правил стилевого оформления кода: это сделает код более «читаемым», поможет в его анализе (и поиске ошибок при необходимости).

При подготовке к промежуточной аттестации необходимо повторить пройденный материал в соответствии с учебной программой и примерным перечнем вопросов. Рекомендуется использовать источники, перечисленные в списке литературы в рабочей программе дисциплины, а также ресурсы электронно-библиотечных систем. Необходимо обратить особое внимание на темы учебных занятий, пропущенных по разным причинам. При необходимости можно обратиться за консультацией и методической помощью к преподавателю.

8. Тематика практических занятий

Занятие №1. Технология решения олимпиадных задач. Алгоритмы. Сложность алгоритмов.

- Основные этапы решения олимпиадных задач.
- Способы ввода-вывода. Ограничения на ресурсы.
- Понятие сложности алгоритма. Оценки сложности алгоритмов.

Занятие №2. Алгоритмы сортировки.

- Задача сортировки массивов данных.
- Виды сортировок. Эффективность различных способов сортировки.
- Быстрая сортировка. Пирамидальная сортировка. Сортировка Шелла.

Занятие №3. Алгоритмы на графах.

- Элементы теории графов в олимпиадной информатике.
- Способы организации представления данных с использованием графов.
- Обход вершин графа. Поиск в глубину. Поиск в ширину. Задача коммивояжера.

9. Методические рекомендации по выполнению лабораторной работы по основам решения олимпиадных задач

При решении олимпиадных задач важно, чтобы ваша программа верно решала задачу за время, не превышающее заданные ограничения.

Тестирование программ

Проверка решения олимпиадной задачи, как правило, заключается в запуске программы-решения, присланной участником, но специальном наборе тестов, подготовленном жюри. Тесты (т.е. наборы входных данных) подобраны таким образом, чтобы охватывать все возможные «случаи» решения конкретной задачи.

Прежде чем отправлять решение на проверку, важно предварительно самостоятельно протестировать решение максимально полно: постаравшись рассмотреть все возможные случаи входных данных.

Особенно важно про тестировании решения рассмотреть так называемые «граничные» случаи: очень большие значения, значения на границах разрешённых интервалов входных типов, «нулевые» значения и т.д.

Высокоточное измерение времени работы программы

Даже если программа-решение будет выдавать верный ответ на любой набор входных данных во время олимпиады такое решение может быть не засчитано, если программа будет работать слишком долго. Поэтому так важно использовать эффективные алгоритмы для решения задач.

Для оценки эффективности алгоритмов обычно используются асимптотические оценки, однако иногда полезно сравнить скорость работы алгоритмов на практике.

Рассмотрим один из наиболее точных способов измерить скорость выполнения кода на языке FreePascal.

Следующий код позволяет узнать время выполнения какого-либо алгоритма в миллисекундах.

```
program Time;
uses
   Windows;
var
   TStart, TFin, TFreq: Int64;

begin
   QueryPerformanceFrequency(TFreq);
   QueryPerformanceCounter(TStart);

   {Здесь реализация алгоритма}

   QueryPerformanceCounter(TFin);
   Writeln('Время: ', (TFin - TStart) / TFreq * 1000:5:5, ' мс');
end.
```