



Развитие функциональной грамотности при решении олимпиадных задач по программированию



Учитель информатики:
Гнедов Андрей Александрович

2022



План доклада

1. Функциональная грамотность
2. Конечная цель
3. Мотивация учащихся
4. Содержание курса
5. Методика проведения занятий





Олимпиадное программирование - это тренировка применения знаний и навыков для решения жизненных задач

1. Как этим управлять?
2. Что и как применить
в новой ситуации





Конечная цель – горящие глаза

1. Развитие интереса к предмету

2. Количественная оценка:

1. Сколько детей дошло до конца учебного года
2. Сколько выпускников школы выбрало IT-специальность
3. Результаты на олимпиадах



1. Поступление в ВУЗ без вступительных испытаний

1. Самоутверждение: я могу это сделать

2. Радость от решённой задачи

3. Уважение сверстников





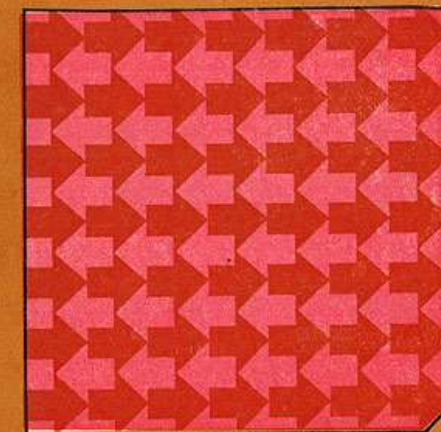
Содержание курса

1. То, что входит в ЕГЭ
2. Стандартные приёмы работы
3. Встроенные структуры данных
4. Типовые алгоритмы

**МАТЕМАТИЧЕСКОЕ
ОБЕСПЕЧЕНИЕ
ЭВМ**

Н. Вирт

**АЛГОРИТМЫ+
СТРУКТУРЫ ДАННЫХ=
ПРОГРАММЫ**





Стандартные приёмы работы

- 1.Копилка
- 2.Рекорд и чемпион
- 3.Флажок да/нет
- 4.Вспомогательные массивы
- 5.Предподсчёт
- 6.Полный перебор

Алгоритмизация и программирование. Язык Python. 10 класс

160

Обработка строк

Полная программа:

```
Fin=open ( "input.txt" )
Fout=open ( "output.txt", "w" )
while True:
    s=Fin.readline()
    if not s: break
```

Алгоритмизация и программирование. Язык Python. 11 класс

96

Раскраска вершин

```
ostov=[] # список выбранных ребер
for k in range(N-1):
    minDist=INF # очень большое число
    for i in range(N):
        for j in range(N):
            if col[i] != col[j] and W[i][j]:
                iMin=i
                jMin=j
                minDist=W[i][j]
    ostov.append((iMin,jMin))
    c=col[jMin]
    for i in range(N):
        if col[i]==c:
            col[i]=col[i]+1
```

© К.Ю. Поляков, Е.А. Еришкин, 2012

Число путей: функция

```
def pathCount ( graph, vStart, vEnd, visited ):
    if vStart==vEnd:
        return 1
    visited.append ( vStart )
    count=0
    for v in graph[vStart]:
        if not v in visited:
            count+=pathCount ( graph, v, vEnd, visited )
    visited.pop()
    return count
```

© К.Ю. Поляков, Е.А. Еришкин, 2012

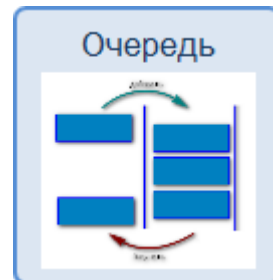
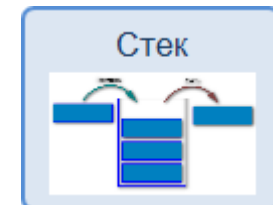
http://profmakelov.spb.ru



Что и как хранить?

Встроенные структуры данных

- 1.Список (вектор)
- 2.Стек и очередь на базе списка
- 3.Словарь
- 4.Множество
- 5.Приоритетная очередь





Типовые алгоритмы

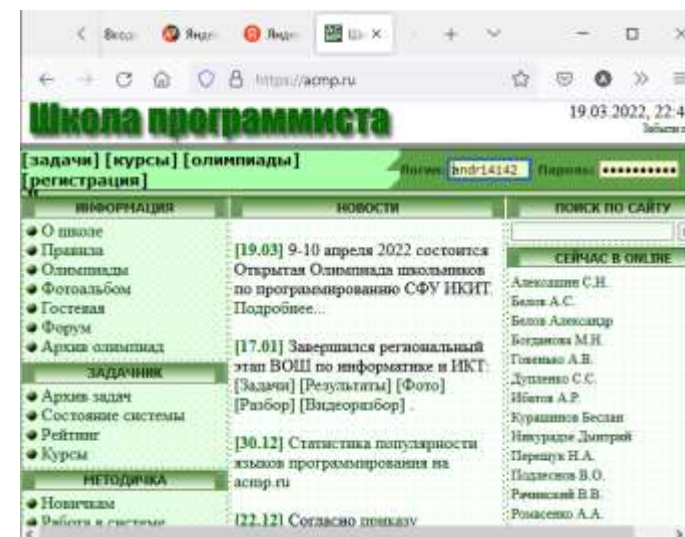
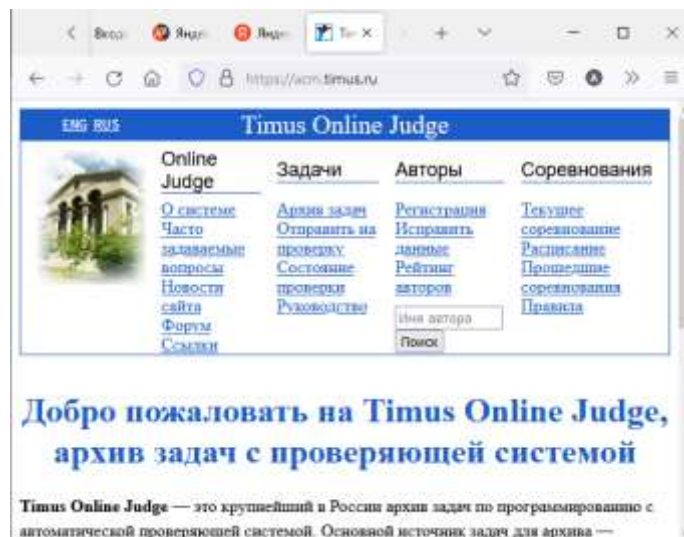
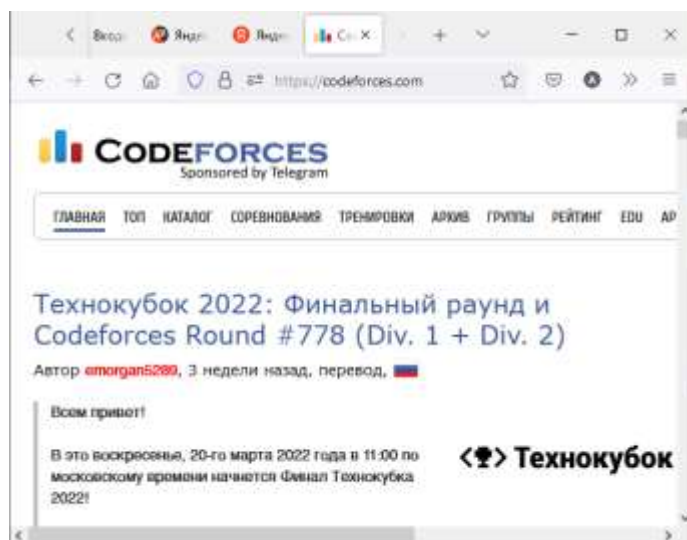
1. Перебор делителей до корня
2. Динамическое программирование
3. Жадные алгоритмы
4. Двоичный поиск
5. Обход дерева





Методика проведения занятий

1. Решение задачи на каждом занятии
2. Сдача решения в онлайн-систему проверки





Сравнение онлайн-систем проверки

СВОЙСТВО	Codeforces.com	Acm.timus.ru	Асmp.ru
Сложность задач	Для студентов	Для студентов	Для школьников
Возможность чтения чужих решений	Есть	Нет	Нет
Структурирование по темам	Не полное	Не полное	Хорошее
Регулярные контесты	1-2 раза в неделю	Нет	Раз в месяц

Разбор методов и структур данных – асmp.ru

Тренировка к олимпиадам – codeforces.com



Спасибо за внимание

Вопросы?

E-mail: GnedovAA@610.ru