

3. Сафин, Р. Р. Исследование совмещенной сушки-пропитки массивных капиллярно-пористых коллоидных материалов / Р. Р. Сафин, Р. Г. Сафин, Н. Р. Галяветдинов, Р. М. Иманаев // Вест. казан. гос. техн. ун-та. – 2006. – №6. – С.78 – 85.

4. Свойства термодревесины // Режим доступа: <http://www.lestermo.ru/termodrevesina.html>. – Дата доступа: 01.04.2019.

5. Термодревесина: изготовление, свойства, применение // Режим доступа: <http://derevo.ua/articles/details/termodrevesina-izgotovlenie-svoystvaprimenenie-11>. – Дата доступа: 01.04.2019.

6. Термодревесина: технологии, применение, перспектива // Режим доступа: <http://www.derevo.ru/publikatsii/50-tekhnika-tekhnologii/sushilnoeproduzvodstvo/1591-termodrevesina-tekhnologiya-primeneniaperspektivy.html>. – Дата доступа: 01.04.2019.

7. Термодревесина // Режим доступа: <http://dry-tree.ru/termodrevesina/>. – Дата доступа: 01.04.2019.

8. Шубин, Г. С. Сушка и тепловая обработка древесины / Г. С. Шубин. – М.: Лесн. пром-сть, 1990. – 336 с.

УДК 452

Балашкова Е. М.

КОНТРОЛЬ ПОТВОРЕНИЯ В ГЕНЕРАЦИИ ЧИСЛОВОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

БНТУ, г. Минск

Научный руководитель: канд. техн. наук, доцент Дробыш А. А.

Часто преподаватели сталкиваются с ситуацией, в которой необходимо распределить имеющиеся варианты заданий определенному количеству студентов.

Такая ситуация напоминает генерацию числовой последовательности, элементами которой являются варианты заданий, а номер этого элемента отражает номер обучающегося в списке студентов. Среди элементов могут встречаться и члены, имеющие одинаковые значения. Следует заметить, чтобы приблизить число списываний студентами заданий друг у друга к минимуму нужно раздать одинаковых вариантов как можно меньше.

Рассматривая данную проблему, была поставлена цель – разработать алгоритм генерации числовой, ограниченной с обеих сторон по-

следовательности из определенного количества элементов с минимальным числом повторений элементов этой последовательности.

Пусть дана задача, согласно которой необходимо создать программу, генерирующую числовую последовательность из N целых чисел, лежащих в диапазоне от n до k , и свести повторения элементов этой последовательности к минимуму.

В программе число элементов последовательности отражает переменная `numberOfElements` и значения нижней `lowlimit`, верхней `uplimit` границ допустимых значений.

Для хранения самой числовой последовательности используется такая структура данных, как целочисленный массив `masElement`. Количество элементов массива хранит переменная `numberOfElements`. Элементы массива отражают элементы числовой последовательности. Номер элемента массива представляет собой порядковый номер члена числовой последовательности.

Для выполнения поставленной задачи создается функция `GenerationArray()`, в которую передаются массив `masElement` и число элементов `numberOfElements` в нем, нижняя `lowlimit` и верхняя `uplimit` границы. Именно в этой функции будет происходить генерация числовой последовательности.

Так как основной задачей является минимизация повторений в числовой последовательности, то необходимо определить максимальное число повторений одного числа и не превышать его. В ходе разработки было выявлено, что максимальное число повторений `maxIteration` одного числа зависит от количества элементов массива и количества чисел, входящих в отрезок `[lowlimit, uplimit]` и задано формулой:

$$\text{maxIteration} = \frac{\text{numberOfElements}}{\text{uplimit} - \text{lowlimit} + 1}$$

где `maxIteration` округляется до наибольшего значения.

Так же следует учитывать `maxNumOfIteration`, отвечающее за то, какое количество чисел могут повторяться максимальное число раз. Это число задается формулой (остаток от деления):

$$\text{maxNumOfIteration} = \frac{\text{numberOfElements}}{\text{uplimit} - \text{lowlimit} + 1}$$

Во втором массиве `helpIteration` контролируются повторения каждого числа из отрезка `[lowlimit, uplimit]`. Количество элементов `helpIteration` соответствует количеству чисел, входящих в отрезок. Элементы `helpIteration` на начальном этапе равны нулю, что означает отсутствие повторений.

Массив `masElement` с помощью цикла заполняется случайными числами с помощью функции `rand()`, которая возвращает целое число из отрезка `[lowlimit, uplimit]`. Если количество повторов данного числа меньше максимального числа повторений `maxIteration`, то рассматриваемому элементу массива `masElement[i]` присваивается сгенерированное случайное число. При невыполнении данного условия требуется заново сгенерировать случайное число.

Так же учитывается факт того, что после выполненного присвоения элементу массива `masElement[i]` сгенерированного случайного числа повтор числового сгенерированного значения может быть равен максимальному числу повторений. В таком случае обращается внимание на количество элементов `iteration`, достигнувших максимального числа повторов. Если `iteration` равно `maxNumOfIteration` (максимальное количество чисел, которые могут повторяться `maxIteration` – максимальное число раз), то `maxNumOfIteration` необходимо уменьшить на единицу.

Убедиться в работоспособности алгоритма можно на примере. Разрабатывается программа, которая генерирует последовательность из 14 элементов, лежащих в отрезке `[5,7]`. В функцию `GenerationArray()` передается массив `masElement` с количеством элементов `numberOfElements=14`, значения нижней границей `lowlimit=5` и верхней границей `uplimit=7`.

```
Введем интервал чисел, которые можем использовать для генерации массива
Нижняя граница: 5
Верхняя граница: 7
Введем размерность массива: 14
Сгенерированная числовая последовательность имеет вид
5 7 7 6 7 5 7 5 6 5 6 6 5 6
```

Рисунок 1 – Генерация последовательности из 14 элементов, лежащих в отрезке `[5, 7]`

Согласно алгоритму вычисляется `maxNumOfIteration` и значение максимального числа повторений `maxIteration` одного числа, которые равны 2 и 5 соответственно (т.е. только два числа из отрезка могут повторяться 5 раз). Далее с учетом всех вычислений генери-

руется последовательность. Результат работы программы приведены на рисунке 1.

Следовательно, можно говорить о том, что алгоритм работает верно и цель достигнута.

УДК 378

Балашкова Е. М.

НЕГАТИВНОЕ ВОЗДЕЙСТВИЕ КОМПЬЮТЕРА НА ЗДОРОВЬЕ ЧЕЛОВЕКА И СПОСОБЫ ЗАЩИТЫ

БНТУ, г. Минск

Научный руководитель: докт. техн. наук, доцент Азаров С. М.

Компьютер стал необходимым устройством в жизни современного человека. Благодаря компьютеру и выходу в интернет мы развлекаемся, общаемся, узнаем информацию, приобретаем новые знания и расширяем свой кругозор.

Основная масса людей соприкасается с компьютером незначительно или вообще не пользуется им. Эти люди правы в том, что для них компьютер абсолютно безвреден.

Однако люди, постоянно работающие с компьютером, отмечают, что иногда через короткое время после начала работы появляются головная боль, болезненные ощущения в области мышц лица и шеи, ноющие боли в позвоночнике, резь в глазах, слезоточивость, нарушение четкого видения, боли при движении рук.

Студенты вынуждены проводить время за экраном персонального компьютера намного больше. Именно компьютер с доступом в Интернет в современном студентам дает доступ к информации, которая необходима для подготовки ответа на вопросы к семинару, написания рефератов, курсовых работ и качественного выполнения лабораторных работ. Так же многие студенты проводят свободное от учебы время играя в игры или общаясь с родственниками и друзьями в социальных сетях.

Такое времяпровождение негативно сказывается на состоянии здоровья молодежи. Однако, большинство специалистов полагает, что при разумном использовании польза от использования компьютера превышает вред.

Компьютер оказывает влияние на: