Задание №1 Оценка средней трудоёмкости алгоритма

**Цель выполнения задания −** овладеть практическими навыками оценки средней трудоемкости алгоритмов методом Марковских цепей и сетевым методом.

**Постановка задачи**

Для алгоритма, разработанного в лабораторной работе №1 и оформленного в виде блок-схемы необходимо:

* Всем блокам содержательной блок-схемы алгоритма назначить трудоёмкости, исходя из конкретных действий, соответствующих этим блокам;
* Вычислить вероятности переходов между блоками в содержательной блок-схеме алгоритма, исходя из соответствующих логических условий;
* Построить граф-схему алгоритма;
* Оценить среднюю трудоемкость заданного алгоритма методом Марковских цепей и сетевым методом;

Представить результаты в следующем виде:

* содержательная блок-схема алгоритма;
* граф-схема алгоритма;
* последовательные преобразования граф-схемы алгоритма;
* результаты оценки эффективности алгоритма двумя методами.

**Задача №1**

Для целочисленного массива (a1, a2,…, a60) определить, является ли сумма его элементов четным числом, и вывести на печать ДА или НЕТ.

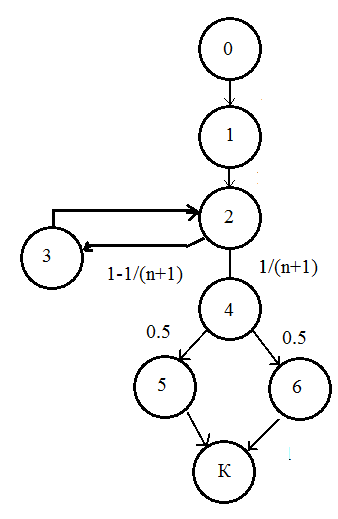
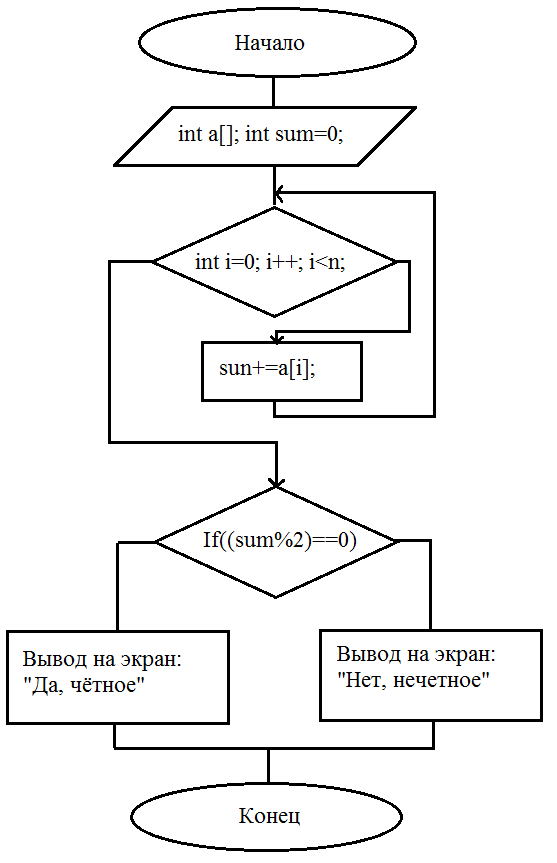


Рисунок 1.- Схема работы алгоритма к задаче 1

***Решение задачи 1***

Проведём оценку трудоёмкостей узлов и вероятностей переходов между ними, анализируя действия, назначенные одноименному блоку в соответствующей блок-схеме.

Трудоёмкости начального и конечного узлов равны нулю т.к. в соответствующих блоках блок-схемы не предусматривается никаких действий, требующих временных затрат.

**Трудоёмкость узла 1**: Θ1=n+1, т.к. в блоке 1 производиться ввод значения, а также ввод последовательности чисел длиной n. Предполагаем, что каждая операция присваивания (ввода) требует одну условную. Единицу времени для реализации.

**Трудоёмкость узла 2**: Θ2=2 т.к. в блоке 2 производятся два действия: увеличение значения параметра цикла i на единицу и сравнение его с конечным значением этого параметра.

Операции сравнения и декремента требуют для своего выполнения по одной условной временной единице каждая.

**Трудоёмкость узла 3**: Θ3=1 т.к. в блоке 3 производиться одно действие: прибавление элемента цикла к переменной.

**Трудоёмкость узла 4**: Θ4=1 т.к. в блоке 4 производиться операция сравнения которая требует для своего выполнения одну временную единицу.

**Трудоёмкость узла 5,6**: одинаковая и равна Θ6=1 т.к. производиться вывод заданного сообщения, которая требует для своего выполнения одну временную единицу.

В соответствии с графом алгоритма вероятность переходов между вершинами равны 1, кроме переходов между вершинами: (2-3) (2-4) (4-5) и (4-6).

Для изучения вероятностей переходов P2-3, P2-4,P4-5 и P4-6 рассмотрим возможные исходы из узла 2 и 4.

2: Цикл с параметром i предполагает выполнение (n+1) раз изменения параметров i и сравнение его значения со значением n, поэтому число возможных исходов в этом случае равно (n+1). Частота выхода из цикла оценивается как 1/(n+1), т.е. P2-4=1/(n+1), а поскольку событие: выполнение тела цикла и выход из цикла составляют полную группу событий то P2-3=1-P2-4=1-1/(n+1).

4: условие чётности предполагает, что вероятности перехода равны: P4-5=0.5 и P4-6=0.5.

Оценка средней трудоёмкости алгоритма методом Марковских цепей.

Составим систему линейных алгебраических уравнений для определения средних числе ni (i=1,2,…,n) пребывания вычисленного процесса в соответствующих состояниях (узлах граф-схемы):

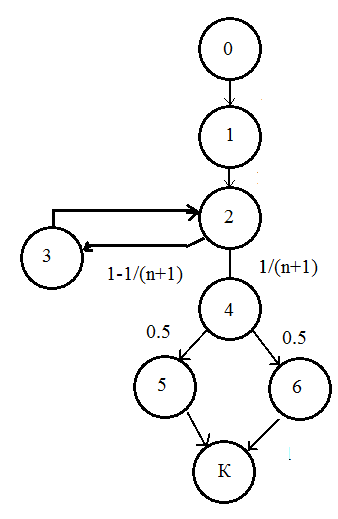
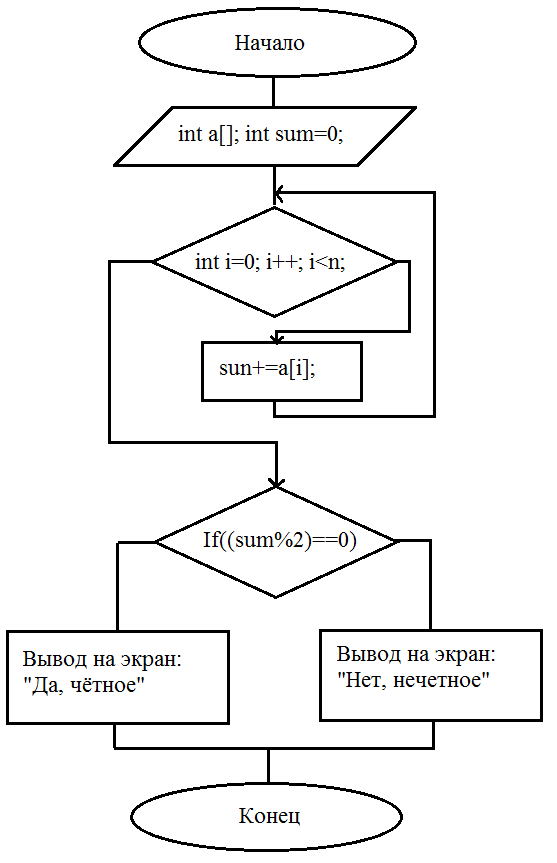
Решением полученной системы уравнений являются значения:

n1=1, n2=1+n, n3=n n­4=1 n5=0.5n6=0.5

Таким образом, средняя трудоёмкость алгоритма как функция от длины последовательности n равна:

Θср==(n+1)n1+2n2+2n3

**Оценка средней трудоёмкости алгоритма сетевым методом.**



Предварительный этап оценки трудоёмкости блоков и оценки вероятностей переходов между ними уже проведён.

Решение задачи.

В соответствующей граф-схеме алгоритма можно выделить цикл:

Рассмотрим цикл, включающий в себя вершины 2-3 и относящийся к 2-му типу цикла.

Вершина 3 соответствуют вершине C для типичной схемы цикла второго типа, вершина 2 – вершине Y, определяющей выход из цикла. Вероятность выхода из цикла P=1/(n+1)

Трудоёмкость вершины Y вычислим следующим образом. Цикл включает в себе 2 вершины. Нужно подсчитать трудоёмкость операций в этих двух вершинах.

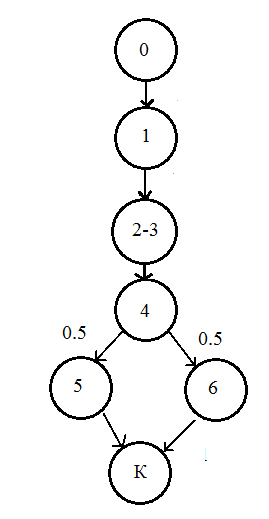
В результате несложных расчётов по формуле получим:

θц2=θсnс+θyny=θy/р+θc(1/р−1)=(θc+ θy)/р−θc

θц2 =(2+ 1)/р−1=(3(n+1))-1

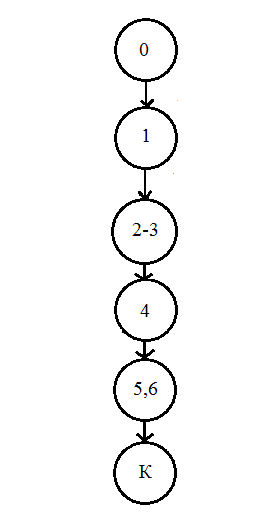
Вершины 2 и 3 в исходном графе можно заменить одной вершиной (2-3).

Преобразованный граф имеет вид:



Вершины 5, 6 в графе можно заменить одной вершиной (5,6).

Преобразованный граф не имеет циклов и представляет собой сеть:



В итоге получим среднюю трудоёмкость алгоритма:

θср=θ1+θ2-7+θ8=(3(n+1))+2=3n+5.

**Задание №2. Оценка маргинальных трудоёмкостей алгоритма**

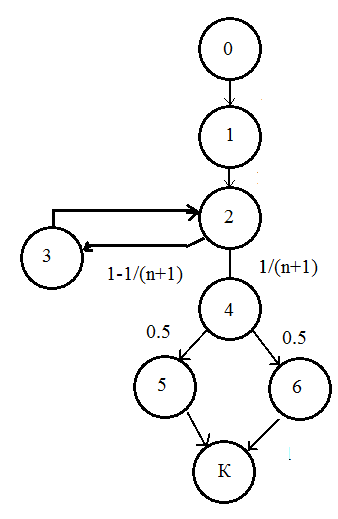
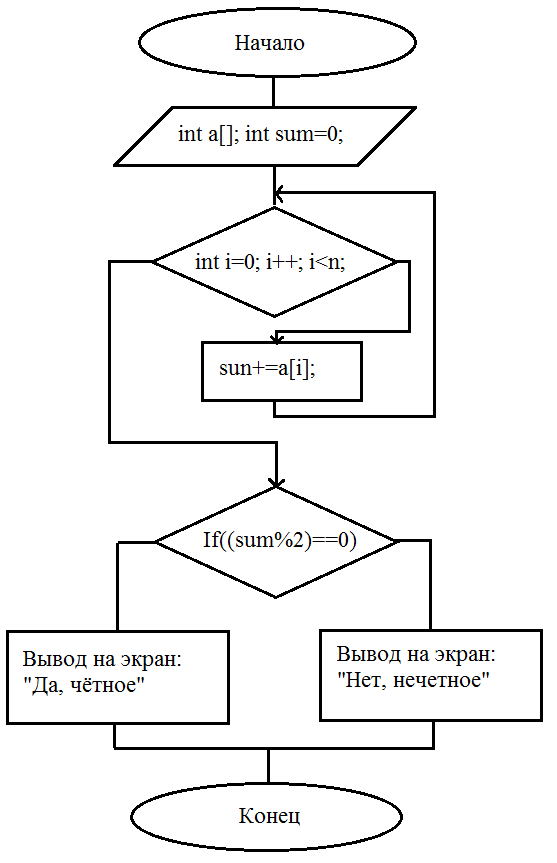
**Цель выполнения задания** − овладеть практическими навыками оценки маргинальных трудоемкостей алгоритмов.

**Постановка задачи**

Для графовой модели алгоритма, разработанной при выполнении задания №1 настоящей лабораторной работы необходимо:

* Оценить минимальную трудоемкость алгоритма;
* Оценить максимальную трудоемкость алгоритма
* Сравнить полученные результаты, с результатами, полученными в предыдущем задании
* На числовой оси отметить все оценки, полученные для заданного алгоритма.
* Минимальную и максимальную трудоёмкости каждого из узлов граф-схемы следует взять равными средней трудоёмкости узла.

Оценка минимальной трудоёмкости алгоритма



В условии задачи требуется оценить минимальную и максимальную трудоёмкости алгоритма. Предполагается, что средние, минимальные и максимальные трудоёмкости отдельных узлов равные между собой. т.е. θi=θimin=θimax.

В граф-схеме можно выделить цикл типа Ц2:

Используя формулу

θц2min=θymin

θц2max=(θcmax+θymax) ⋅ (2/P−1) ,

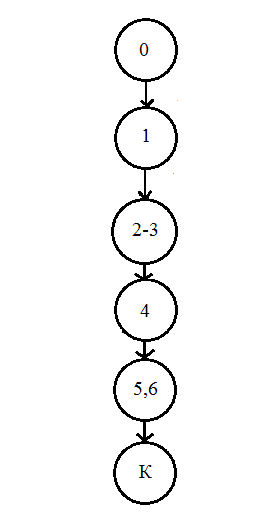
получим:

θц2min=θymin=1,

θц2max=(n+(n+1))\*(2/P-1)=4n2+2n.

После этого цикл заменяем вершину ( ) с соответствующими маргинальныvb трудоёмкостями

Граф-схема алгоритма после этой замены примет вид:



В итоге получим маргинальные трудоёмкости алгоритма:

θmin=θ1min+θ(2-3)min+θ4min +θ(5,6)min=1+1+1+0.5=3.5,

θmax=θ1max+θ(2-3)max+θ4min+θ(5,6)max=1+(4n2+2n)+1+0.5=4n2+2n+2.5

То есть, случайная величина, определяющая трудоёмкость алгоритма, распределена в диапазоне [3.5;4n2+2n+2.5] со средним значением 3n+5.