

Министерство образования и науки Украины
Севастопольский национальный технический университет



МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторной работы
«Математический анализ эффективности
нерекурсивных алгоритмов»
по дисциплине «Алгоритмизация и формальные преобразования»
для студентов направления подготовки
«Компьютерная инженерия»
дневной формы обучения

Севастополь
2014

УДК 004.421

Методические указания к выполнению лабораторной работы «Математический анализ эффективности нерекурсивных алгоритмов» по дисциплине «Алгоритмизация и формальные преобразования» для студентов направления подготовки «Компьютерная инженерия» дневной формы обучения. / Сост. Козлова Е.В. – Севастополь: Изд-во СевНТУ, 2014. – 16с.

Целью методических указаний является оказание помощи студентам специальности 7.091501 – «Компьютерные системы и сети» дневной формы обучения, в изучении основ алгоритмизации при решении инженерных задач.

В настоящих методических указаниях приведены некоторые теоретические сведения, касающиеся основных методов математического анализа нерекурсивных алгоритмов, примеры решения типовых задач, варианты индивидуальных заданий к выполнению лабораторной работы, сформулированы цели выполнения заданий и основные требования по оформлению работы.

Методические рекомендации рассмотрены и утверждены на заседании кафедры кибернетики и вычислительной техники, протокол № 11 от 25 июня 2014 г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензенты: Апраксин Ю.К., д. т. н., профессор,
Щепин Ю.Н., к. т. н., доцент.

СОДЕРЖАНИЕ

Введение	3
1 Некоторые теоретические сведения	4
2 Задание № 1. Оценка средней трудоёмкости алгоритма	9
3 Задание № 2. Оценка маргинальных трудоёмкостей алгоритма	9
4 Примеры решения типовых задач	10
5 Контрольные вопросы	16
Библиографический список	16

Введение

Целью выполнения данной лабораторной работы является освоение студентами методов математического анализ эффективности алгоритмов и овладение навыками вычисления средней, минимальной и максимальной трудоёмкостей алгоритмов.

Номер варианта индивидуального задания для выполнения лабораторной работы соответствует номеру фамилии студента в списке учебной группы.

В рамках лабораторной работы должно быть выполнено два задания, которые предполагают получение средних и маргинальных оценок трудоёмкости алгоритма, разработанного при выполнении задания в лабораторной работе №1.

Работа должна быть выполнена **аккуратно**, желательно в виде компьютерного набора. Однако не запрещается выполнять рисунки и схемы вручную, но обязательно с использованием линейки.

Разрешается выполнять расчеты как вручную, так и с использованием любых средств автоматизации вычислений (могут быть использованы любые пакеты прикладных программ). Соответствующие листинги должны быть внедрены в текст контрольной работы как «объект» с **дополнительными пояснениями**. На титульном листе обязательно должны присутствовать номер выполняемого варианта. Работа должна включать следующие разделы:

1. Содержание
2. Задание №1 Оценка средней трудоёмкости алгоритма
 - 2.1. Постановка задачи
 - 2.2. Оценка средней трудоёмкости алгоритма методом марковских цепей
 - 2.3. Оценка средней трудоёмкости алгоритма сетевым методом
3. Задание №2. Оценка маргинальных трудоёмкостей алгоритма
 - 3.1. Постановка задачи
 - 3.2. Оценка минимальной трудоёмкости алгоритма
 - 3.3. Оценка максимальной трудоёмкости алгоритма
4. Анализ полученных результатов

Работы, выполненные небрежно или не для своего варианта задания будут возвращены без проверки!!!

1. Некоторые теоретические сведения

Для конкретной реализации алгоритма, содержащего циклы или условные переходы, трудоемкость Θ – случайная величина. Оценками этой случайной величины могут служить: $\Theta_{\text{ср}}$ – среднее значение трудоемкости, а также Θ_{min} и Θ_{max} , минимальное и максимальное значения трудоемкости, соответственно.

Оценку величины $\Theta_{\text{ср}}$ полезно производить тогда, когда речь идет о массовом (многократном) применении алгоритма.

Однако если речь идет о малом или даже однократном применении алгоритма, более пригодны оценки Θ_{min} и Θ_{max} .

Хорошей моделью для оценки трудоемкости алгоритма является общепринятая запись алгоритма в виде некоторой блок-схемы (БСА). Однако, с нашей точки зрения содержательный смысл блоков и условных переходов несущественны, а важна трудоемкость блоков и вероятность переходов между ними.

Рассмотрим, как с помощью этой модели можно оценить интересующие нас характеристики. Начнем с оценки $\Theta = \Theta_{\text{ср}}$.

В общем случае БСА можно представить в виде графа G с $(K+2)$ -мя вершинами, где 0 и K соответствуют началу и концу алгоритма, а остальные K вершин – блокам БСА. Каждая вершина i характеризуется трудоемкостью Θ_i ее выполнения. Переходы между блоками i и j обозначены в модели дугами (i,j) . Дуга в графе характеризуется вероятностью перехода P_{ij} в процессе реализации алгоритма.

Будем считать P_{ij} постоянными и известными, также как и θ_i , $i=1,...,K$.

Естественно, должны выполняться условия:

$$\sum_{i=1}^K p_{ij} = 1, i=1,..., K. \quad (1)$$

Предположим, что нам каким-то образом удалось определить значения величин n_i (n_i – среднее число прохождения блока i в процессе реализации алгоритма).

Тогда величину $\Theta_{\text{ср}}$ легко оценить значением:

$$\Theta_{\text{ср}} = \sum_{i=1}^K \Theta_i \cdot n_i. \quad (2)$$

Методы оценки трудоемкости алгоритмов, в общем случае, разнятся способом вычисления среднего значения числа прохождений i -го блока (n_i).

1.1. Оценка средней трудоемкости алгоритма методом марковских цепей

Оценка средней трудоемкости алгоритма методом марковских цепей предполагает постоянство P_{ij} и независимость этих величин от того, каким образом процесс попал в вершину i .

Процесс реализации алгоритма в этом случае можно рассматривать как марковский процесс с $(K+2)$ -мя состояниями, где состояние «1» является начальным, а состояние «K» – поглощающим.

При этом числа $n_1, \dots, n_K$ можно рассматривать как средние числа пребывания марковского процесса в состояниях 1, 2, ..., K.

В соответствии с теорией марковских цепей, вычислять их можно как корни системы линейных уравнений:

$$\begin{cases} n_1 = 1 + \sum_{j=1}^K p_{j1} \cdot n_j \\ n_i = \sum_{j=1}^K p_{ji} \cdot n_j \end{cases} \quad (3)$$

1.2. Оценка трудоёмкости алгоритма сетевым методом

1.2.1 Сетевой метод оценки трудоёмкости для графа без циклов

Сетевой метод непосредственно применим только к графу без циклов. В таком графе вершины предварительно нумеруются так, чтобы начальная вершина имела наименьший номер, например, 0, а очередной номер i присваивается вершине, в которую входят дуги от уже пронумерованных вершин, с номерами, меньшими i .

Для такого графа величины n_i рассчитываются по формулам:

$$\begin{cases} n_1 = 1 \\ n_i = \sum_{j=1}^K p_{ji} \cdot n_j \end{cases} \quad (4),$$

а средняя трудоёмкость алгоритма соответственно равна

$$\Theta_{cp} = \sum_{i=1}^K \Theta_i \cdot n_i. \quad (5)$$

1.2.2. Сетевой метод оценки алгоритма для графа с циклами

Если граф, для оцениваемого алгоритма содержит циклы, то применяется предварительный метод сжатия циклов и замены их вершинами с эквивалентными трудоёмкостями.

Циклом называется совокупность вершин и дуг, охватываемых дугой обратной связи. Указанная совокупность вершин и дуг называется *телом цикла*. Циклы могут быть многократно вложены друг в друга.

Циклом первого ранга называется цикл, не содержащий внутри себя ни одного цикла. Очевидно, что тело цикла первого ранга – ориентированный граф без циклов.

Циклом второго ранга называется цикл, содержащий хоть бы один цикл первого ранга.

Циклом S-го ранга называется цикл, содержащий внутри себя хотя бы один цикл $(S-1)$ -го ранга и, возможно, циклы меньших рангов.

Процесс сжатия циклов в эквивалентные им по трудоемкости вершины начинается с самых внутренних циклов, т.е. с циклов первого ранга. Поскольку тело такого цикла – ориентированный граф без циклов, то к нему может быть применён сетевой метод оценки трудоемкости.

После такой оценки все вершины этого подграфа, за исключением условной вершины цикла, легко «сжать» в одну вершину «С», присвоив ей трудоемкость θ_{cp} . Трудоемкость условной вершины «Y» обозначается θ_y .

Как показано на рисунке 1, в процессе сжатия циклов первого ранга можно выделить три типа циклов.

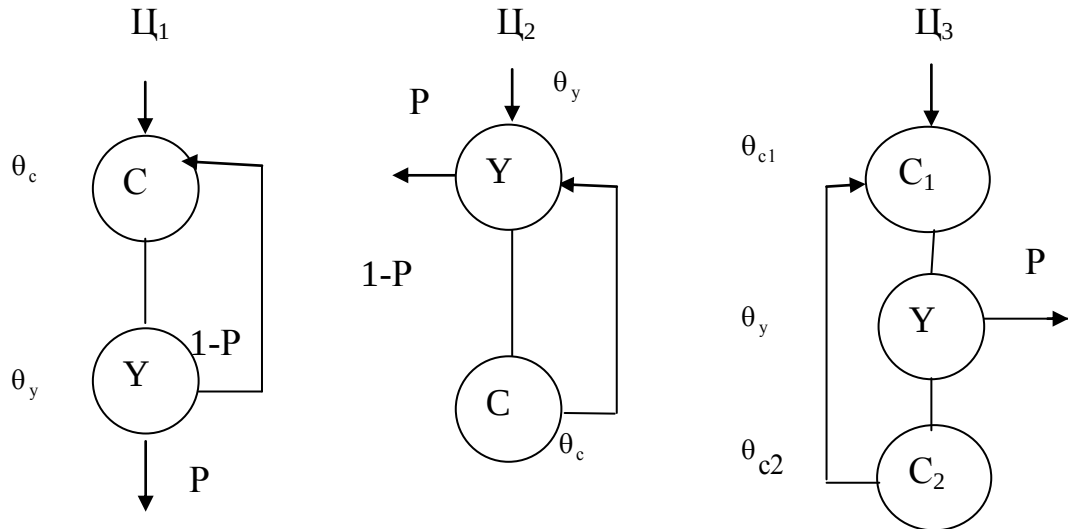


Рисунок 1. Типы циклов, встречающиеся в блок-схемах алгоритмов.

Применяя методику расчетов трудоемкостей методом Марковских цепей, получим следующие формулы для вычисления трудоёмкостей циклов 1-го, 2-го и 3-го типов циклов, соответственно:

$$\theta_{ц1} = \theta_c n + \theta_y n_y = (\theta_c + \theta_y) / p. \quad (6)$$

$$\theta_{ц2} = \theta_c n_c + \theta_y n_y = \theta_y / p + \theta_c (1/p - 1) = (\theta_c + \theta_y) / p - \theta_c. \quad (7)$$

$$\theta_{ц3} = \theta_{c1} n_{c1} + \theta_y n_y + \theta_{c2} n_{c2} = (\theta_{c1} + \theta_y + \theta_{c2}) / p - \theta_{c2}. \quad (8)$$

После сжатия циклов первого ранга, они исчезают, а циклы второго ранга становятся циклами первого ранга, и т.д. Процесс повторяется до тех пор, пока все циклы не будут сжаты и не получен граф без циклов, к которому может быть применен сетевой метод.

1.3. Оценка маргинальных трудоемкостей алгоритма

Маргинальными оценками называются оценки минимальной и максимальной трудоемкости алгоритма.

Задача определения маргинальных оценок ставится следующим образом. Для каждой вершины i граф-схемы алгоритма известны θ_{imin} и θ_{imax} ,

соответственно минимальная и максимальная трудоёмкости вершины, а также P_{ij} – вероятности перехода от вершины i к вершине j ($i=1,2,\dots,K$).

Требуется найти значения величин $A=\theta_{\min}$ и $B=\theta_{\max}$ – минимальной и максимальной, соответственно, трудоёмкостей алгоритма по предлагаемой граф-схеме.

На графе D без циклов определить маргинальные трудоёмкости не составляет особого труда.

Обозначим через A_i и B_i – минимальную и максимальную, соответственно, трудоёмкости реализации алгоритма на момент прохождения блока i .

По определению $A_0=B_0=0$;

$$A_i = \min_{(j,i) \in D} \{A_j\} + \theta_{i\min}, \quad (9)$$

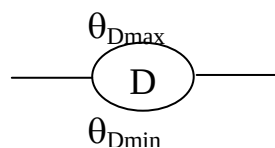
$$B_i = \max_{(j,i) \in D} \{B_j\} + \theta_{i\max}. \quad (10)$$

Дойдя до конечной вершины K , получим:

$$A = A_K = \min_{(j,K) \in D} \{A_j\}, \quad (11)$$

$$B = B_K = \max_{(j,K) \in D} \{B_j\}. \quad (12)$$

После этого граф-схему алгоритма можно заменить одной вершиной с маргинальными трудоёмкостями $\theta_{D\min}$ и $\theta_{D\max}$:



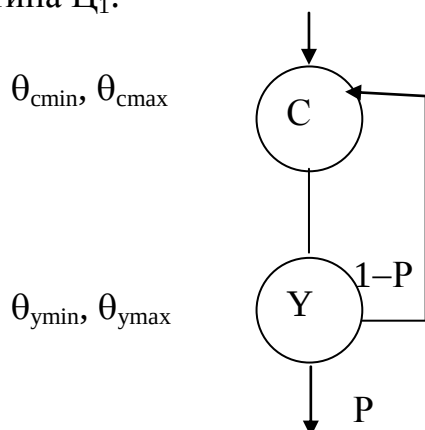
Для графа с циклами так же, как и при вычислении средней трудоёмкости, применяется сетевой метод, причем вершинам, соответствующим сжимаемым циклам присваиваются маргинальные оценки, равные:

$$\theta_{ц\min} = A_c \cdot n_{ц\min} \text{ и } \theta_{ц\max} = B_c \cdot n_{ц\max},$$

где A_c и B_c – соответственно, минимальные и максимальные трудоёмкости тела цикла, а $n_{ц\min}$ и $n_{ц\max}$ – минимальные и максимальные числа прохождения цикла, соответственно.

Определим эти числа для всех типов циклов:

1. Цикл типа $Ц_1$:



Здесь $n_{ц\min} = 1$, $A_c = \theta_{c\min} + \theta_{y\min}$

Чтобы оценить $n_{цmax}$, предположим, что число проходов цикла распределено равномерно на интервале $[n_{цmin}, n_{цmax}]$ со средним $n_{ц}=1/P$.

Тогда, для определения $n_{цmax}$ имеем следующее уравнение:

$$(n_{цmin} + n_{цmax})/2 = n_{ц}.$$

Заменив $n_{ц}$ на $1/P$, получим

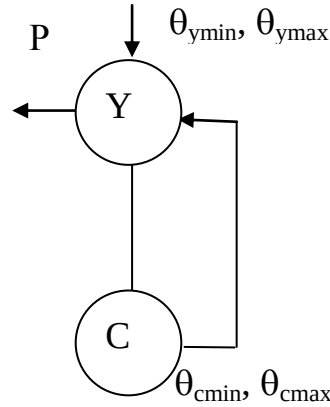
$$(1 + n_{цmax})/2 = 1/P.$$

Отсюда, $n_{цmax} = 2/P - 1$.

Далее, $B_c = \theta_{cmax} + \theta_{ymax}$, так что,

$$\left. \begin{aligned} \theta_{ц1min} &= \theta_{cmin} + \theta_{ymin} \\ \theta_{ц1max} &= (\theta_{cmax} + \theta_{ymax}) \cdot (2/P - 1) \end{aligned} \right\}. \quad (13)$$

2. Цикл типа Ц₂:



Для этого типа цикла $n_{cmin} = 0$, $\theta_{ц2min} = \theta_{ymin}$.

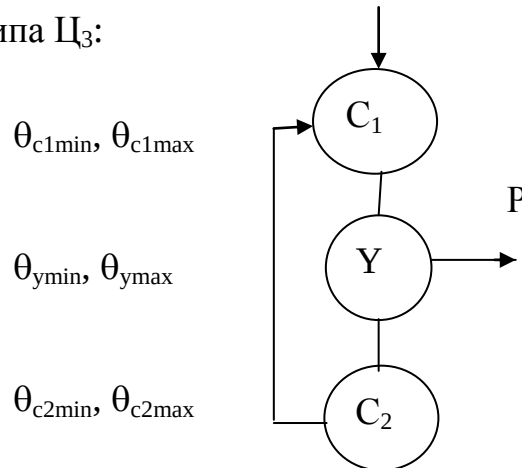
Аналогично случаю, рассмотренному выше, предположим, что число проходов цикла распределено равномерно на интервале $[n_{min}, n_{цmax}]$ со средним $n_{ц} = 1/P - 1$.

Тогда $(n_{цmin} + n_{цmax})/2 = n_{ц}$.

Произведя замену, $n_{цmax}/2 = 1/P - 1$, получим $n_{цmax} = 2/(P - 2)$, так что,

$$\left. \begin{aligned} \theta_{ц2min} &= \theta_{ymin} \\ \theta_{ц2max} &= (\theta_{cmax} + \theta_{ymax}) \cdot (2/P - 1) \end{aligned} \right\}. \quad (14)$$

3. Цикл типа Ц₃:



Действуя аналогично предыдущему, можно показать, что здесь

$$\left. \begin{aligned} \theta_{ц3min} &= \theta_{c1min} + \theta_{ymin} \\ \theta_{ц3max} &= (\theta_{c1max} + \theta_{ymax}) \cdot (2/P - 1) + \theta_{c2max} \cdot (1/P - 2) \end{aligned} \right\} \quad (15).$$

Или иначе:

$$\theta_{ц3max} = 2 \cdot (\theta_{c1max} + \theta_{ymax} + \theta_{c2max}) / P - (\theta_{c1max} + \theta_{ymax} + 2 \cdot \theta_{c2max}).$$

2. Задание № 1. Оценка средней трудоемкости 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:

1. Всем блокам содержательной блок-схемы алгоритма назначить трудоёмкости, исходя из конкретных действий, соответствующих этим блокам;
2. Вычислить вероятности переходов между блоками в содержательной блок-схеме алгоритма, исходя из соответствующих логических условий;
3. Построить граф-схему алгоритма;
4. Оценить среднюю трудоемкость заданного алгоритма методом марковских цепей и сетевым методом;
5. Представить результаты в следующем виде:
 - содержательная блок-схема алгоритма;
 - граф-схема алгоритма;
 - последовательные преобразования граф-схемы алгоритма;
 - результаты оценки эффективности алгоритма двумя методами.

3. Задание № 2. Оценка маргинальных трудоемкостей 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:

1. Оценить минимальную трудоемкость алгоритма;
2. Оценить максимальную трудоемкость алгоритма;
3. Сравнить полученные результаты, с результатами, полученными в предыдущем задании

4. 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.

4. Примеры решения типовых задач

Задача 1. Вычислить среднюю трудоёмкость алгоритма решения следующей задачи: из заданной последовательности чисел $A=\{A_i\}$ ($i=\overline{1,n}$) сформировать новую последовательность чисел $B=\{B_i\}$, где $B_i=(A_i+c)d$, причём c и d – константы. Блок-схема алгоритма и соответствующая ей графовая модель представлены на рисунке 2.

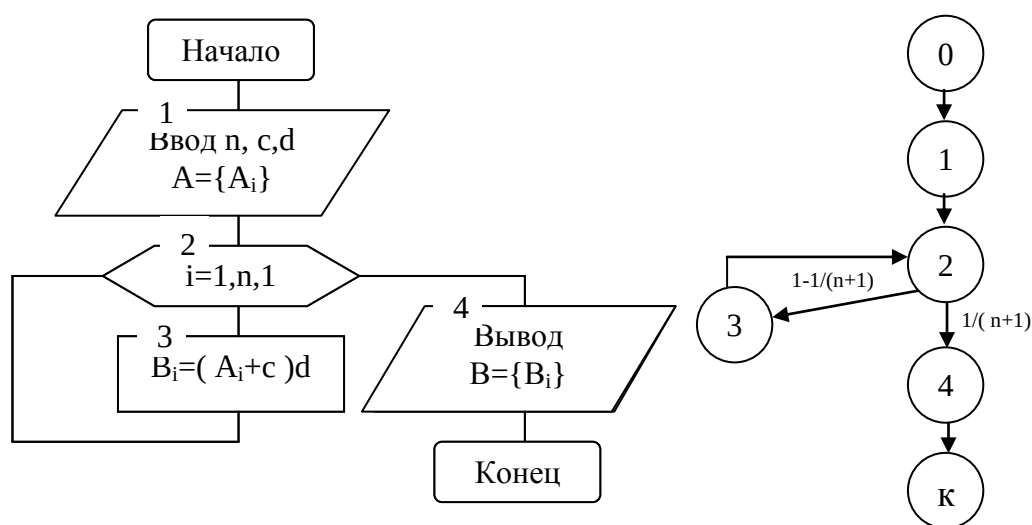


Рисунок 2. Блок-схема алгоритма и соответствующая ей графовая модель.

Решение задачи 1

Проведём оценку трудоёмкостей узлов и вероятностей переходов между ними, анализируя действия, назначенные одноимённому 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 $\Theta_1=n+3$, т.к. в блоке 1 производится ввод трех значений констант (n , c и d), а также ввод последовательности чисел длиной n . Предполагаем, что каждая операция присваивания (ввода) требует одну условную единицу времени для реализации.

Трудоёмкость узла 2 $\Theta_2=2$, 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.

В блоке 3 производится вычисление арифметического выражения $(A_i+c)d$, т.е. получение суммы (A_i+c) , а затем произведения $(A_i+c)d$. Поэтому *трудоемкость узла 3* $\Theta_3=2$.

Вывод последовательности чисел длиной n в блоке 4 определяет трудоемкость соответствующего узла $\Theta_4=n$.

В соответствии с графом алгоритма вероятности переходов между вершинами равны 1, кроме переходов между парами вершин: (2-3) и (2-4).

Для назначения вероятностей переходов $P_{2,3}$ и $P_{2,4}$ рассмотрим возможные исходы из узла 2.

Цикл с параметром i предполагает выполнение $(n+1)$ раз изменения параметра 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. $P_{2,4}=1/(n+1)$, а поскольку события: выполнение тела цикла и выход из цикла составляют полную группу событий то $P_{2,3}=1-P_{2,4}=1-1/(n+1)$.

Решим поставленную задачу методом марковских цепей

Составим систему линейных алгебраических уравнений для определения средних чисел n_i ($i=1, 2, \dots, n$) пребывания вычислительного процесса в соответствующих состояниях (узлах граф-схемы):

$$\begin{cases} n_1 = 1 \\ n_2 = n_1 + n_3 \\ n_3 = \frac{n}{n+1} \cdot n_2 \\ n_4 = \frac{1}{n+1} \cdot n_2 \end{cases}$$

Решением полученной системы уравнений являются значения:

$$n_1=1, n_2=1+n, n_3=n, n_4=1.$$

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

$$\Theta_{cp} = \sum_{i=1}^n \Theta_i \cdot n_i = (n+1) \cdot n_1 + 2 \cdot n_2 + 2 \cdot n_3 + n \cdot n_4 = (n+1) \cdot 1 + 2 \cdot (1+n) + 2n + n \cdot 1 = 4n+3.$$

Задача 2. Требуется вычислить среднюю трудоемкость алгоритма, представленного граф-схемой на рисунке 3, сетевым методом.

В постановке этой задачи содержательное наполнение блоков в блок-схеме алгоритма отсутствует и предполагается, что предварительный этап оценки трудоемкости блоков и оценки вероятностей переходов между ними уже проведён, т. е. построена графовая модель алгоритма. Это лишь облегчает нашу задачу.

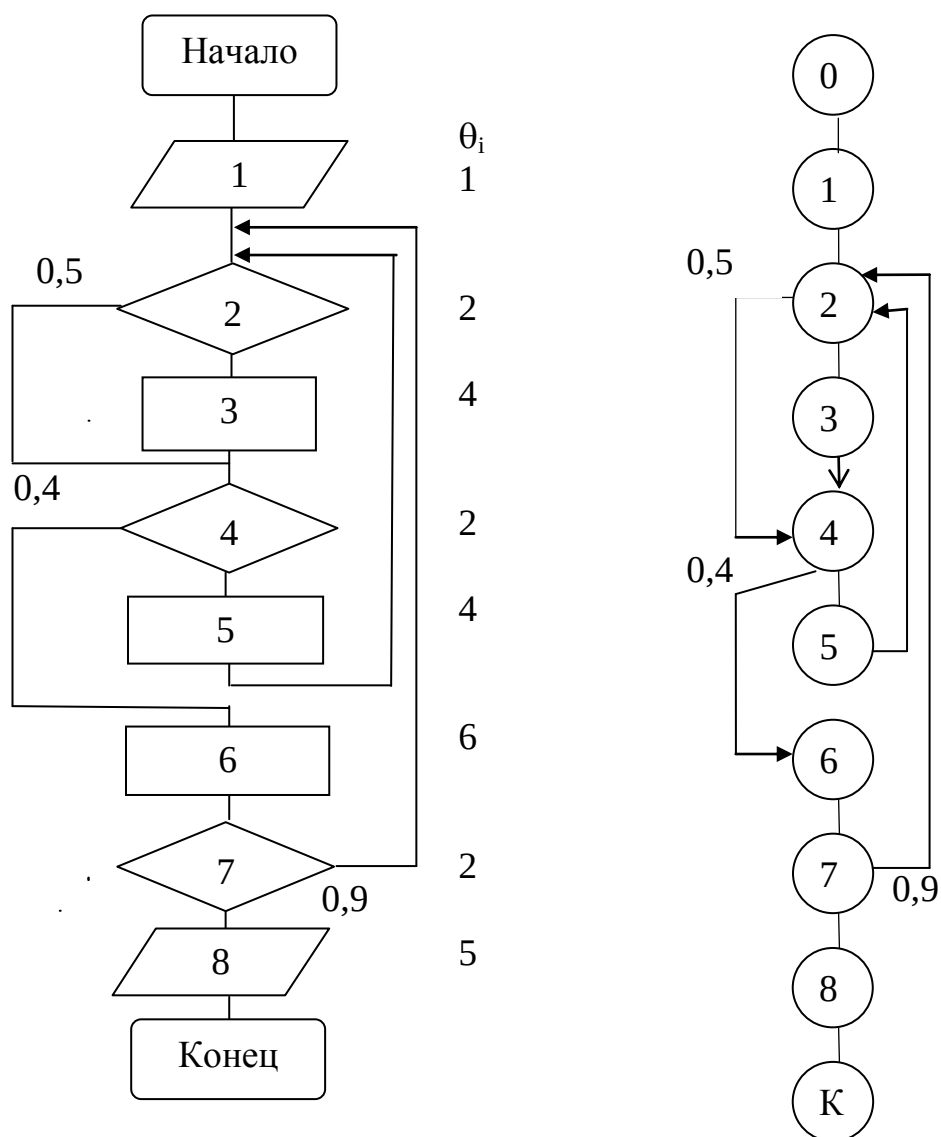
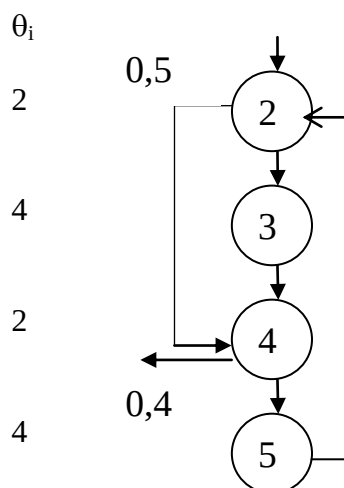


Рисунок 3. Блок-схема алгоритма и соответствующий ей граф переходов.

Решение задачи 2.

В соответствующей граф-схеме алгоритма можно выделить два цикла: цикл 1-го ранга – (2–3–4–5) и цикл 2-го ранга – (2–3–4–5–6–7).

Рассмотрим сначала цикл первого ранга, включающий в себя вершины 2, 3, 4 и 5, и относящийся к третьему типу циклов.



Вершины 2, 3 и 4 соответствуют вершине C_1 для типичной схемы цикла третьего типа, вершина 5 – вершине C_2 , а вершина 4 – условной вершине Y , определяющей выход из цикла. Вероятность выхода из цикла $p=0,4$.

Трудоёмкость вершины C_1 вычислим следующим образом. Подграф, включающий вершины 2, 3 и 4, является сетью, поэтому к нему можно применить сетевой метод расчета трудоёмкости.

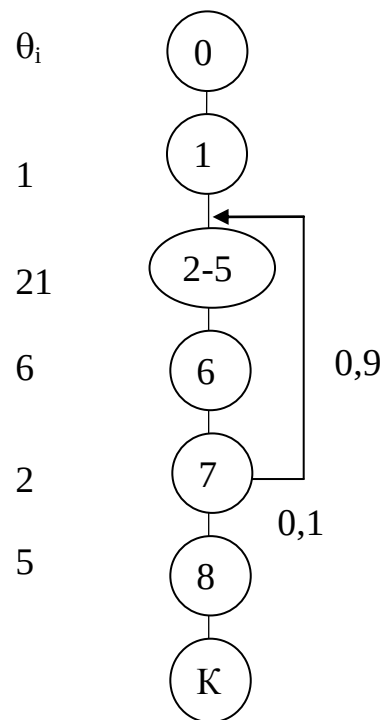
В результате несложных расчётов получим:

$$n_2=1; n_3=0,5 \cdot n_2=0,5; n_4=n_3+0,5 \cdot n_2=1; \theta_{C_1}=2 \cdot 1+4 \cdot 0,5=4.$$

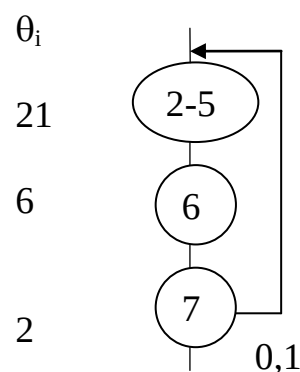
Трудоёмкость вершины $\theta_{C_2}=\theta_5=4$, а трудоёмкость условной вершины $\theta_y=\theta_4=2$. Тогда, согласно формуле (8),

$$\theta_{2-5}=(\theta_{C_1}+\theta_y+\theta_{C_2})/p - \theta_{C_2}=(4+2+4)/0,4 - 4=21.$$

Вершины 2, 3, 4 и 5 в исходном графе можно заменить одной вершиной (2 – 5). Преобразованный граф имеет вид:



В преобразованном графе вершины (2-5), 6 и 7 образуют цикл, который относится к типу 1.



Подграф, включающий вершины (2-5), и 6, является сетью, поэтому к нему можно применить сетевой метод расчета трудоёмкости, т.е.:

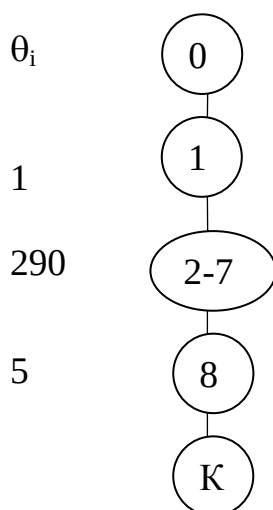
$$n_{2-5}=1; n_6=n_{2-5}=1; \theta_c=21+6=27.$$

Трудоёмкость условной вершины $\theta_y=\theta_7=2$.

Тогда, согласно формуле (6):

$$\theta_{2-7}=(\theta_c+\theta_y)/p=(27+2)/0,1=290.$$

Вершины (2 – 5), 6 и 7 в графе можно заменить одной вершиной (2–7). Преобразованный граф не имеет циклов и представляет собой сеть:

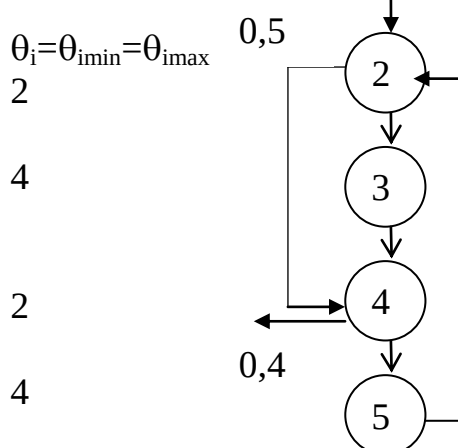


В итоге получим среднюю трудоёмкость алгоритма

$$\theta_{cp}=\theta_1+\theta_{2-7}+\theta_8=1+290+5=296.$$

Задача 3. В условиях задачи 2 требуется оценить минимальную и максимальную трудоёмкости алгоритма. Предполагается, что средние, минимальные и максимальные трудоёмкости отдельных узлов равны между собой, т.е. $\theta_i=\theta_{imin}=\theta_{imax}$.

В этой граф-схеме (рисунок 3) можно выделить цикл типа Ц₃:



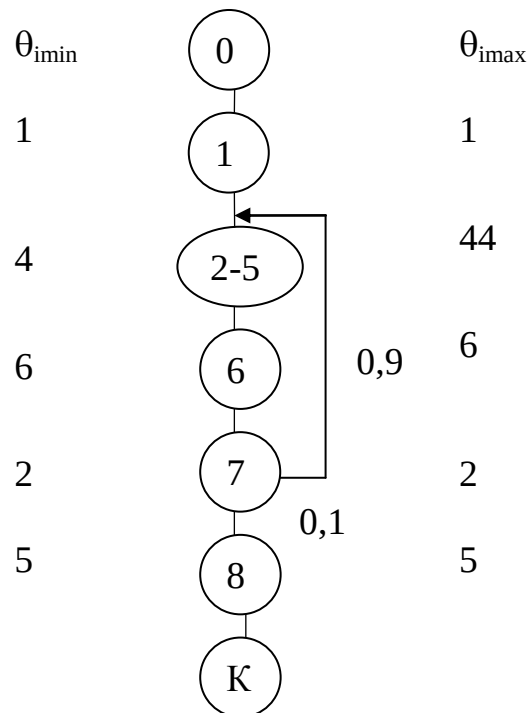
Используя формулы (15), получим:

$$\theta_{ц3min}=\theta_2+\theta_4=2+2=4,$$

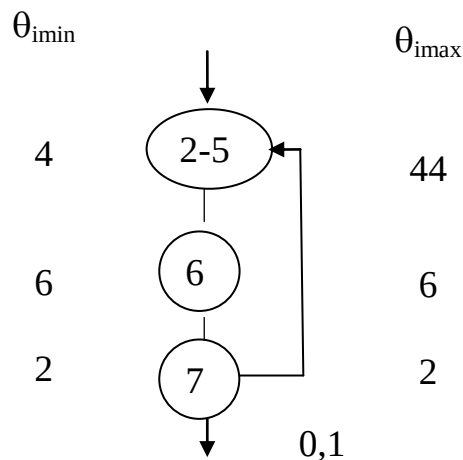
$$\theta_{ц3max}=2\cdot((\theta_2+\theta_3)+\theta_4+\theta_5)/0.4-((\theta_2+\theta_3)+\theta_4+2\theta_5)=2\cdot(6+2+4)/0.4-(6+2+8)=44.$$

После этого цикл заменяем вершиной (2–5) с соответствующими маргинальными трудоёмкостями.

Граф-схема алгоритма после этой замены примет вид:



В преобразованном графе вершины (2–5), 6 и 7 образуют цикл, который относится к типу 1.



Вершины (2–5) и 6 образуют тело C цикла, маргинальные трудоёмкости которого равны:

$$\theta_{Cmin} = \theta_{(2-5)min} + \theta_{6min} = 4 + 6 = 10 \text{ и}$$

$$\theta_{Cmax} = \theta_{(2-5)max} + \theta_{6max} = 44 + 6 = 50.$$

Вершина 7 – условная вершина Y , для которой

$$\theta_{Ymin} = \theta_{Ymax} = 2.$$

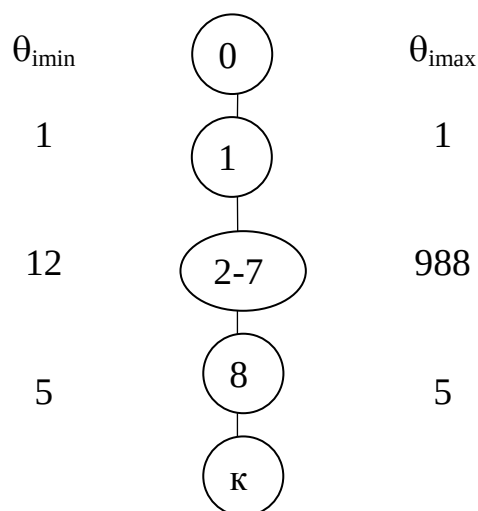
Вероятность выхода из цикла $p = 0,1$.

Используя формулу (13) получим:

$$\theta_{II1min} = \theta_{Cmin} + \theta_{Ymin} = 10 + 2 = 12 \text{ и}$$

$$\theta_{II1max} = (\theta_{Cmax} + \theta_{Ymax}) \cdot (2/p - 1) = (50 + 2) \cdot (2/0.1 - 1) = 52 \cdot 19 = 988.$$

Преобразованная граф-схема алгоритма примет вид:



В итоге получим маргинальные трудоёмкости алгоритма:

$$\theta_{min} = A_K = \theta_{1min} + \theta_{(2-7)min} + \theta_{8min} = 1 + 12 + 5 = 18,$$

$$\theta_{max} = B_K = \theta_{1max} + \theta_{(2-7)max} + \theta_{8max} = 1 + 988 + 5 = 994.$$

То есть, случайная величина, определяющая трудоемкость алгоритма, распределена в диапазоне [18, 994] со средним значением равным 296.

5. Контрольные вопросы

1. Каковы основные критерии оценки эффективности алгоритмов?
2. Что такое трудоемкость алгоритма?
3. Какие модели применяются для проведения оценки алгоритмов?
4. Какие виды трудоемкостей алгоритмов Вы знаете?
5. В каких случаях отдаётся предпочтение оценке средней трудоемкости алгоритма, а в каких минимальной и максимальной?
6. В чём состоит основная идея оценки средней трудоемкости алгоритма сетевым методом?
7. Перечислите достоинства и недостатки оценки средней трудоемкости алгоритма методом марковских цепей и сетевым методом.

Библиографический список

1. Ахо А.В. Хопкрофт Дж. Э., Ульман Дж. Д. Структуры данных и алгоритмы.—М.: Издательский дом «Вильямс», 2000.—950с.
2. Грин Д. Кнут Д. Математические методы анализа алгоритмов.—М.: Мир, 1987.—590с.
3. Кормен Т. Алгоритмы построение и анализ (второе издание)—М.: Издательский дом «Вильямс», 2007.—633с