

**Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря
Сікорського»
Інститут прикладного системного аналізу**

А.Є.Коваленко

ПОБУДОВА КОДІВ НА ОСНОВІ ТИПОВИХ АЛГОРИТМІВ КОДУВАННЯ ДАНИХ

Методичні вказівки із самостійної роботи студентів з дисципліни «Теорія
інформації і кодування»

Затверджено Вченою радою Інституту прикладного системного аналізу
”НТУУ “КПІ ім. Ігоря Сікорського” як методичні вказівки із самостійної
роботи для студентів, які навчаються за спеціальністю “Системний аналіз”

**Київ
НТУУ «КПІ імені Ігоря Сікорського»
2017**

УДК 004.02(075.8)

Гриф надано Вченовою радою Інституту прикладного системного аналізу НТУУ “КПІ імені Ігоря Сікорського” (Протокол № 6 від 27.06 2017 р.)

Рецензент:

Ю.О.Кулаков, д-р. техн. наук, проф.. (Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»).

Відповідальний
редактор

В.Д.Романенко, д-р.техн.наук, проф.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського »

Коваленко А.Є. Побудова кодів на основі типових алгоритмів кодування даних : методичні вказівки із самостійної роботи для студентів з дисципліни «Теорія інформації і кодування» підготовки бакалаврів за спеціальністю “Системний аналіз” / Уклад. А.Є.Коваленко.-2-ге вид., розшир. та доповн.- К.: ПСА НТУУ «КПІ імені Ігоря Сікорського», 2017.- 151 с.

Системно подано вказівки для самостійної роботи студентів з дисципліни «Теорія інформації і кодування», приклади розв'язання задач з різних розділів курсу для аудиторної і самостійної роботи студентів.

Методичні вказівки містять рекомендації про можливий порядок вивчення і закріплення матеріалу курсу, вимоги на різних етапах навчання, типові приклади, необхідні для розв'язання і оформлення задач індивідуальних завдань, питання і варіанти завдань для проведення модульної контрольної роботи та залікової контрольної роботи кредитного модуля курса, варіанти індивідуальних завдань для студентів і приклади їх оформлення.

Призначено для студентів підготовки бакалаврів за спеціальністю “Системний аналіз”

УДК 004.02(075.8)

За редакцією укладача

©А.Є.Коваленко, 2017

ЗМІСТ

| | |
|---|-----------|
| 1 ПОРЯДОК ВИКОНАННЯ СТУДЕНТОМ САМОСТІЙНОЇ РОБОТИ | 5 |
| 1.1 Планування самостійної роботи | 5 |
| 1.2 Підготовка до практичних занять | 6 |
| 1.3 Підготовка до контрольних робіт | 7 |
| 2 ОСНОВНІ ПОНЯТТЯ | 8 |
| 2.1 Оцінка інформації джерела повідомлень | 8 |
| 2.1.1 Теорія інформації і кодування | 8 |
| 2.1.2 Інформаційна ентропія, кількість інформації, код | 8 |
| 2.2 Оцінка кількості інформації дискретного каналу | 10 |
| 2.2.1 Умовна і сумісна ентропії каналу | 10 |
| 2.2.2 Часткова і загальна умовна ентропія | 12 |
| 2.2.3 Статистична надлишковість повідомлень | 13 |
| 2.2.4 Блокове кодування | 13 |
| 2.2.5 Продуктивність джерела | 14 |
| 2.3 Приклади розв'язування задач | 15 |
| 2.4 Задачі та завдання | 17 |
| 3 ОСНОВИ ПОБУДОВИ КОДІВ | 18 |
| 3.1 Класифікація кодів | 18 |
| 3.2 Коди і алгоритми стиснення даних | 18 |
| 3.3 Завадостійкі коди і алгоритми їх побудови | 18 |
| 4 АЛГОРИТМИ СТИСНЕННЯ ДАНИХ | 19 |
| 4.1 Загальна характеристика | 19 |
| 4.2 Кодові таблиці і кодові дерева | 19 |
| 4.3 Алгоритм кодування за методом Шеннона – Фано | 20 |
| 4.4 Алгоритм кодування за методом Хаффмана | 21 |
| 4.5 Алгоритм блокового кодування за методом Хаффмана | 22 |
| 4.6 Алгоритм арифметичного кодування | 26 |
| 4.7 Задачі та завдання з кодів Шеннона–Фано і Хаффмана | 30 |
| 4.8 Адаптивний алгоритм Хаффмана з упорядкованим деревом | 30 |
| 4.9 Задачі та завдання за адаптивним алгоритмом Хаффмана | 38 |
| 4.10 Словникові алгоритми стиснення даних | 38 |
| 4.10.1 Основні поняття | 38 |
| 4.10.2 Алгоритм LZ77 | 39 |
| 4.10.3 Алгоритм LZSS | 41 |
| 4.10.4 Алгоритм LZ78 | 43 |
| 4.10.5 Алгоритм LZW | 44 |
| 4.11 Задачі та завдання за словниковими алгоритмами | 46 |
| 5 АЛГОРИТМИ ПОБУДОВИ ЗАВАДОСТІЙКИХ КОДІВ | 47 |
| 5.1 Лінійні блокові коди | 47 |
| 5.2 Ітеративний код | 48 |
| 5.3 Способи подання лінійних кодів | 50 |
| 5.4 Перевірна матриця і кодовий синдром | 52 |
| 5.5 Поліноміальне кодування інформації. Циклічні коди | 55 |
| 5.6 Задачі та завдання | 61 |
| 6 ОСНОВНА ЛІТЕРАТУРА | 62 |
| 7 ДОДАТКОВА ЛІТЕРАТУРА | 62 |
| ДОДАТОК А ПРИКЛАДИ ПИТАНЬ І ВАРИАНТІВ ЗАВДАНЬ ДО КОНТРОЛЬНОЇ РОБОТИ 1 | 63 |
| ДОДАТОК Б ПРИКЛАДИ ПИТАНЬ І ВАРИАНТІВ ЗАВДАНЬ ДО КОНТРОЛЬНОЇ РОБОТИ 2 | 64 |
| ДОДАТОК В ПРИКЛАДИ ПИТАНЬ І ВАРИАНТІВ ЗАВДАНЬ ДО ЗАЛІКОВОЇ КОНТРОЛЬНОЇ РОБОТИ ... | 66 |
| ДОДАТОК Г ПРИКЛАДИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ | 69 |
| ДОДАТОК Д ПРИКЛАДИ ОФОРМЛЕННЯ ТИТУЛЬНОЇ СТОРІНКИ ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ | 86 |
| ДОДАТОК Ж ПРИКЛАДИ ОФОРМЛЕННЯ ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ | 87 |

ВСТУП

Побудова кодів на основі типових алгоритмів кодування даних є однією з провідних цілей вивчення дисципліни «Теорія інформації і кодування». Різноманіття алгоритмів кодування та складність їх теоретичного обґрунтування ускладнює самостійну роботу студента і вимагає поступового системного практичного засвоєння і застосування цих алгоритмів.

Самостійну роботу студентів варто розглядати на різних етапах навчання: планування роботи з врахуванням рейтингу, виконання індивідуальних завдань та контрольних робіт, підготовки до заліку.

Змістовне наповнення самостійної роботи включає ознайомлення студентів з можливим переліком питань та з варіантами завдань, які виносять на контрольні роботи та залік, обмежений теоретичний матеріал для розв'язання задач, приклади типових навчальних задач та їх оформлення, задачі для самостійного опрацювання тощо.

Порядок застосування методичних вказівок залежить від цілей і рівня підготовленості студента. Можна запропонувати такі схеми послідовності використання рекомендацій за нумерованими розділами, підрозділами, пунктами, підпунктами, зміст яких може мінятись з часом. Планування самостійної роботи і попереднє ознайомлення з дисципліною: 1.1, 6, 7.

Підготовка до першої атестації: 1.1, 1.2, 2, 6, Додаток А.

Підготовка до першої контрольної роботи: 2, 3.1, Додаток А.

Підготовка до другої атестації: 1.2, 3, 4, 6, 7, Додаток Б.

Підготовка до другої контрольної роботи: 2, 3, 4 Додаток Б.

Підготовка до заліку: 1.2, 1.3, 2, 3, 4, 5, 6, 7, Додаток В.

Приклади індивідуальних завдань наведені у додатку Г

Приклади оформлення виконання індивідуального завдання наведені у додатку Ж

1 ПОРЯДОК ВИКОНАННЯ СТУДЕНТОМ САМОСТІЙНОЇ РОБОТИ

1.1 Планування самостійної роботи

Організація самостійної роботи студентів спирається на планування виконання необхідних дій підготовки до практичних занять, контрольних робіт протягом семестру і підготовки до заліку. Значне місце у самостійній роботі займають питання практичного засвоєння матеріалу дисципліни і подальшого використання отриманих знань для розв'язання задач, зокрема на контрольних роботах, заліку. Самостійна робота включає виконання індивідуальних завдань та підготовку доповідей за результатами виконання на практичних заняттях.

Така робота повинна проводитись у рамках рейтингової системи оцінювання успішності навчання.

Самостійна робота студента полягає у своєчасному виконанні навчального графіка з отриманням відповідних балів. Отримані бали RD є визначальними на першій і другій атестаціях.

Для отримання «зараховано» з першої проміжної атестації (восьмий тиждень) студент повинен мати не менше ніж 19 балів.

Для отримання «зараховано» з другої проміжної атестації (14-й тиждень) студент повинен мати не менше ніж 36 балів.

За отриманим рейтингом $RD \geq =60$ балів студент має можливість отримати залікову оцінку (залік) «автоматом». У разі відмови від отримання оцінки «автоматом» попередній рейтинг скасовується і студент виконує залікову контрольну роботу, за якою визначають оцінку.

За умови $60 > RD \geq =40$ студент виконує залікову контрольну роботу. Після виконання залікової роботи попередній рейтинг скасовується.

1.2 Підготовка до практичних занять

Активна робота студента на практичних заняттях передбачає можливість отримання студентом високих балів, залежно від виконання індивідуальних завдань.

Виступ на практичному занятті подається як захист результату виконання індивідуального завдання. Тому самостійна робота студента повинна враховувати особливості оформлення, яке спрощує розуміння правильності і повноти виконання завдання. Тобто повинна бути відображенна послідовність виконання окремих етапів розв'язання задачі (завдання) для спрощення і спрошення можливої перевірки.

Перелік вимог до оформлення результатів формулює викладач при визначенні індивідуальних завдань, а характер вимог залежать від особливостей лекційного матеріалу і загальноприйнятих правил пошуку і обробки даних за завданнями.

Основними цілями курсу «Теорія інформації і кодування» є отримання знань з алгоритмізації обробки даних і практичне застосування алгоритмів стисненні даних і побудови завадостійких кодів. Тому захист результатів виконання індивідуального завдання на практичному занятті передбачає вміння студента чітко пояснити зміст алгоритму побудови коду, кодових слів і володіння студентом навичками і уміннями у застосуванні цих алгоритмів (зокрема, програмування окремих фрагментів, обробка контрольних даних тощо).

1.3 Підготовка до контрольних робіт

Робочою навчальною програмою з дисципліни передбачено протягом семестру виконання однієї модульної контрольної роботи, яку поділена на дві контрольні роботи тривалістю по одній академічній годині. Ваговий показник контрольної роботи складає 20 балів. Тому максимальна кількість балів по двох контрольних роботах становить $20 \text{ балів} \times 2 = 40 \text{ балів}$.

Критерії оцінювання для кожної роботи визначаються складністю задач і завдань в межах сумарного вагового балу.

Варіанти контрольних робіт відповідають тематиці лекційного курсу та практичних занять. Форму проведення контрольної роботи (зокрема, у письмовому вигляді, за тестами тощо) визначає викладач за тиждень до її проведення. Приклади контрольних питань та приклади варіантів завдань до першої і другої контрольної роботи наведені у додатках А, Б. Склад цих питань на поточний навчальний рік визначають не менше ніж за тиждень до проведення контрольної роботи.

Зазвичай контрольні роботи передбачають теоретичні питання і практичні задачі з поточних тем лекційного курсу і практичних занять. Тому склад варіантів індивідуальних завдань контрольних робіт визначають за переліком питань і типовими задачами практичних занять.

2 ОСНОВНІ ПОНЯТТЯ

2.1 Оцінка інформації джерела повідомлень

2.1.1 Теорія інформації і кодування

Теорія інформації (ТІ) — це розділи математики та кібернетики, які досліджують процеси зберігання, перетворення і передачі інформації. Теорія інформації ґрунтується на теорії ймовірностей і математичній статистиці. Базові положення ТІ пов'язані з вимірюванням і оцінкою кількості інформації, інформаційною ентропією, комунікаційними системами, криптографією, корекцією помилок і іншими важливими областями.

Теорія кодування (ТК) вивчає властивості кодів та їх придатність для розв'язання специфічних задач у різних галузях науки і техніки і ґрунтуються на результатах ТІ. Зокрема ТК застосовують при побудові комунікаційних систем, систем захисту інформації, стиснення даних, криптографії, для знаходження і виправлення помилок, при мережевому кодуванні даних.

Важливими напрямами теорії кодування є стиснення даних і кодування даних інформаційних джерел (source coding) та канальне кодування (forward error correction, channel coding). Основною метою кодування джерела є стиснення даних для їх ефективнішої передачі. Канальне кодування забезпечує завадостійкість кодів.

2.1.2 Інформаційна ентропія, кількість інформації, код

Інформаційна ентропія є мірою невизначеності випадкової величини. За Шенноном її визначають як абсолютну межу найкращого стиснення даних без втрат.

Повідомлення розглядають як послідовність незалежних та однаково розподілених випадкових величин. Випадкові величини на скінченному алфавіті називають дискретними випадковими величинами (ДВВ).

Кількість інформації джерела дискретних повідомлень x_i , оцінюють середньою кількістю інформації, що припадає на одне елементарне повідомлення, як ентропією джерела за формулою

$$H(X) = - \sum_{i=1}^k p_i \log_2 p_i , \quad i=1\dots k ,$$

де k - об'єм алфавіту джерела для незалежних повідомлень,

та кількість інформації, що міститься в ДВВ X щодо ДВВ Y (взаємна інформація) за формулою

$$I(X,Y) = \sum_{ij} p_{ij} \log_2 \frac{p_{ij}}{p_i q_j}$$

де p_i, q_j – ймовірності, що визначають закони розподу ймовірностей ДВВ ($P(X=X_i)=p_i, P(Y=Y_j)=q_j$);

p_{ij} - ймовірність, що визначає розподіл сумісних ймовірностей системи $P(X=X_i, Y=Y_j)=p_{ij}$.

Кодування полягає у перетворенні інформації на впорядкований набір символів, елементів, знаків. При кодуванні кожному повідомленню з деякої множини, що називається ансамблем повідомлень, ставиться у відповідність кодова комбінація – набір символів (елементів, знаків). Множина повідомлень називається алфавітом повідомлень, або первинним алфавітом, а множина символів (елементів, знаків) називається алфавітом джерела, або вторинним алфавітом.

Побудовану відповідно до певної схеми кодування множину кодових комбінацій називають кодом.

Для нерівномовірних елементарних повідомлень x_i ентропія зменшується, що оцінюють статистичною надлишковістю

$$\rho_x = 1 - \frac{H(X)}{H(X)_{\max}} = 1 - \frac{H(X)}{\log_2 k},$$

де $H(X)$ - ентропія джерела повідомлень;

$H(X)_{\max} = \log_2 k$ - максимально досяжна ентропія даного джерела

2.2 Оцінка кількості інформації дискретного каналу

2.2.1 Умовна і сумісна ентропії каналу

Ентропію джерела з k статистично незалежних повідомлень $H(x)$ називають безумовною ентропією.

Для статистично залежних повідомлень: переданого в канал повідомлення x і отриманого з каналу повідомлення y застосовують умовну ймовірність $p(x/y)$ появі повідомлення x_i за умови, що вже вибрано повідомлення y_j або умовну ймовірність появі повідомлення y_j , якщо вже отримане повідомлення x_i $p(y/x)$. У загальному випадку $p(x/y) \neq p(y/x)$.

Сумісна ймовірність $p(x,y)$ ($p(x,y) = p(x) \cdot p(y/x)$, $p(x,y) = p(y) \cdot p(x/y)$) дозволяє обчислити умовні ймовірності $p(y/x) = p(x,y)/p(x)$, $p(x/y) = p(x,y)/p(y)$. Зокрема, для статистично незалежних повідомлень x, y : $p(y/x) = p(y)$, $p(x/y) = p(x)$.

Статистичну залежність джерела Y від джерела X задають матрицею прямих переходів повідомлень x_i ($i=1\dots k$) джерела X в повідомлення y_j ($j=1\dots k$) джерела Y :

| X | Y | | | | | |
|----------------|------------------------------------|------------------------------------|-----|------------------------------------|-----|------------------------------------|
| | Y ₁ | Y ₂ | ... | y _i | ... | y _k |
| x ₁ | p(y ₁ /x ₁) | p(y ₂ /x ₁) | ... | p(y _j /x ₁) | ... | p(y _k /x ₁) |
| x ₂ | p(y ₁ /x ₂) | p(y ₂ /x ₂) | ... | p(y _j /x ₂) | ... | p(y _k /x ₂) |
| ... | ... | ... | ... | ... | ... | ... |
| x _i | p(y ₁ /x _i) | p(y ₂ /x _i) | ... | p(y _j /x _i) | ... | p(y _k /x _i) |
| ... | ... | ... | ... | ... | ... | ... |
| x _k | p(y ₁ /x _k) | p(y ₂ /x _k) | ... | p(y _j /x _k) | ... | p(y _k /x _k) |

Правильний вибір джерелом Y повідомлень X відповідає головній діагоналі цієї матриці $x_1 \rightarrow y_1, x_2 \rightarrow y_2, \dots, x_k \rightarrow y_k$, і умові нормування

$$\sum_j p(y_j/x_i) = 1, \quad i=1..k$$

Статистична залежність джерела X від джерела Y подається матрицею зворотних переходів типу $x_i \leftarrow y_j$ з умовних ймовірностей $p(x_i/y_j)$ за умови нормування $\sum_i p(x_i/y_j) = 1, \quad j=1..k$

| X | Y | | | | | |
|----------------|------------------------------------|------------------------------------|----|------------------------------------|-----|------------------------------------|
| | y ₁ | Y ₂ | .. | y _i | ... | y _k |
| x ₁ | p(x ₁ /y ₁) | p(x ₁ /y ₂) | .. | p(x ₁ /y _j) | ... | p(x ₁ /y _k) |
| x ₂ | p(x ₂ /y ₁) | p(x ₂ /y ₂) | .. | p(x ₂ /y _j) | ... | p(x ₂ /y _k) |
| ... | | | .. | ... | ... | ... |
| x _i | p(x _i /y ₁) | p(x _i /y ₂) | .. | p(x _i /y _j) | ... | p(x _i /y _k) |
| ... | | ... | .. | ... | ... | ... |
| x _k | p(x _k /y ₁) | p(x _k /y ₂) | .. | p(x _k /y _j) | ... | p(x _k /y _k) |

За безумовною імовірностю $P(X)=\{p(x_i)\}$ знаходять сумісні імовірності

$$p(X, Y) = \begin{bmatrix} p(x_1, y_1) & p(x_1, y_2) & \dots & p(x_1, y_k) \\ p(x_2, y_1) & p(x_2, y_2) & \dots & p(x_2, y_k) \\ \dots & \dots & \dots & \dots \\ p(x_k, y_1) & p(x_k, y_2) & \dots & p(x_k, y_k) \end{bmatrix}.$$

За згорткою по і дістанемо ряд розподілу безумовних ймовірностей

$$P(Y)=\{p(y_j)\}, j=1\dots k: \quad p(y_j) = \sum_{i=1}^k p(x_i, y_j), \quad j = 1 \dots k, \quad \text{а за згорткою по } j -$$

розподіл $P(X)=\{p(x_i)\}, i=1\dots k: \quad p(x_i) = \sum_{j=1}^k p(x_i, y_j), \quad i = 1 \dots k$

2.2.2 Часткова і загальна умовна ентропія

Часткова умовна ентропія - це кількість інформації на одне повідомлення джерела X за умови джерелом Y повідомлення y_j ,

$H(X/y_j) = -\sum_i p(x_i/y_j) \log_2 p(x_i/y_j), j=1\dots l$, або кількість інформації на одне

повідомлення джерела Y за відомим станом джерела X

$H(Y/x_i) = -\sum_j p(y_j/x_i) \log_2 p(y_j/x_i), i=1\dots k$, де $X=\{x_1, x_2, \dots, x_k\}$,

$Y=\{y_1, y_2, \dots, y_l\}$ - алфавіти повідомень; x_i - повідомлення X , щодо якого визначається $H(Y/x_i)$ алфавіту Y за умови вибору X повідомлення x_i ; y_j - повідомлення Y , щодо якого визначається $H(X/y_j)$ алфавіту X за умови вибору y_j ; i, j - номер повідомлення з алфавітів X ; Y ; $p(x_i/y_j)$, $p(y_j/x_i)$ – умовні ймовірності.

Загальні умовні ентропії визначають як :

$$H(X/Y) = \sum_j p(y_j) H(X/y_j),$$

$$H(Y/X) = \sum_i p(x_i) H(Y/x_i).$$

або, після підстановки значень, як

$$H(X/Y) = -\sum_j p(y_j) \sum_i p(x_i/y_j) \log_2 p(x_i/y_j) =$$

$$= -\sum_j \sum_i p(x_i, y_j) \log_2 p(x_i/y_j),$$

$$H(Y/X) = -\sum_i p(x_i) \sum_j p(y_j/x_i) \log_2 p(y_j/x_i) = \\ = -\sum_i \sum_j p(x_i, y_j) \log_2 p(y_j/x_i),$$

де $p(x_i, y_j)$ - сумісна імовірність x_i, y_j , а $p(x_i/y_j), p(y_j/x_i)$ – їх умовні імовірності.

2.2.3 Статистична надлишковість повідомлень

Статистичну надлишковість за наявною статистичною залежністю повідомлень визначають як $\rho_{X/Y} = 1 - H(X/Y)/H(X)$, де $H(X/Y)$ - загальна умовна ентропія джерела X стосовно джерела Y ; $H(X)$ - ентропія джерела X .

Загальну статистичну надлишковість алфавіту джерела інформації визначають як $\rho_{X,Y} = \rho_X + \rho_{X/Y} - \rho_X \rho_{X/Y}$. За малих значень $\rho_X, \rho_{X/Y}$ $\rho_{X,Y} = \rho_X + \rho_{X/Y}$.

2.2.4 Блокове кодування

Кодування інформації з метою зменшення її надлишковості, називають ефективним, або статистичним. Для цього застосовують укрупнення (об'єднання) повідомлень у блоки (блокування). Оскільки імовірності зв'язків між блоками менші ніж між повідомленнями, то і менша залежність між блоками.

Ентропію об'єднання двох джерел $H(X,Y)$ (взаємна ентропія) як середню кількість інформації на два будь-які повідомлення джерел X і Y визначають за $H(X,Y) = -\sum_{ij} p(x_i, y_j) \log p(x_i, y_j)$.

Оскільки $p(x_i, y_j) = p(y_j, x_i)$, то $H(X, Y) = H(Y, X)$ і $H(X, Y) = H(Y) + H(X/Y)$.

Звідси $H(Y/X) = H(X, Y) - H(X)$, $H(X/Y) = H(X, Y) - H(Y)$.

Кількість інформації, передане каналом зв'язку джерелом X спостерігачу Y за наявності завад і статистичного взаємозв'язку X і Y знаходять за формулою $I(X, Y) = H(Y) + H(X) - H(X, Y) = H(X) - H(X/Y) = H(Y) - H(Y/X)$.

2.2.5 Продуктивність джерела

Продуктивність джерела щодо конкретного повідомлення визначають як

$$V_{\text{дж}_i} = \frac{I(X_i)}{\tau_i}, \text{ де } \tau_i \text{ — проміжок часу вибору повідомлення } x_i.$$

Продуктивність джерела інформації характеризують середнім значенням

$$V_{\text{дж}} = \frac{1}{\tau_{\text{ср}}} \sum_{i=1}^k p_i I(X_i) = \frac{H(X)}{\tau_{\text{ср}}},$$

де $\tau_{\text{ср}}$ — середній час вибору джерелом одного повідомлення.

Повідомлення x_i передається по каналу зв'язку спостерігачеві Y , роль якого відіграє приймальний пристрій. Вибір повідомлень $y_j \in Y$ джерелом Y характеризує процес передачі інформації каналом зв'язку від джерела X на вихід джерела Y . При цьому взаємна кількість інформації $I(X, Y)$ - це середня кількість інформації про стан джерела X , що міститься в одному повідомленні джерела Y .

Швидкість передачі інформації каналом зв'язку знаходять за формулою

$$V = \frac{I(X, Y)}{\tau}, \text{ де } I(X, Y) \text{ - це середня кількість інформації про стан джерела}$$

X в одному повідомленні джерела Y . τ , - час вибору одного повідомлення.

За відсутніх інформаційних втрат (завад в каналі) $I(X, Y) = H(X) = H(Y) = H(X, Y)$, а за високим рівнем завад $H(X/Y) = H(X)$, $H(Y/X) = H(Y)$, $H(X, Y) = H(X) + H(Y)$ і $I(X, Y) = 0$.

Завади спотворюють передані повідомлення і умовна ентропія змінюється в межах: $0 \leq H(X/Y) \leq H(X)$, $0 \leq H(Y/X) \leq H(Y)$. Тому кількість переданої по каналу зв'язку інформації визначають як $I(X, Y)=H(X)-H(X/Y)$.

Максимально можлива швидкість передачі інформації по каналу називається пропускною здатністю, або ємністю каналу зв'язку С

$$C = \frac{1}{\tau} [I(X, Y)]_{\max} = \frac{1}{\tau} [H(X) - H(X/Y)]_{\max}.$$

За рівноімовірних повідомлень пропускна здатність за відсутності завад

$$C = \frac{1}{\tau} H(X)_{\max} = \frac{1}{\tau} \log_2 k .$$

За наявних завад умовна пропускна здатність каналу дорівнює

$$C = \frac{1}{\tau} [\log_2 k - H(X/Y)] .$$

При зменшенні рівня завад пропускна здатність каналу С прямує до максимального значення, а при збільшенні рівня завад – до нуля.

2.3 Приклади розв'язування задач

Приклад 2.1 Знайти ентропії ДВВ X, Y, Z та кількість інформації, що містить ДВВ $Z=|X-Y|$ стосовно X . X, Y – незалежні ДВВ., які задаються розподілами ймовірностей (табл. 2.5, табл. 2.6)

Таблиця 2.1- Розподіл ймовірностей системи ДВВ X

| | | | | |
|---|-----|-----|-----|-----|
| X | 1 | 2 | 3 | 4 |
| p | 1/8 | 1/8 | 1/4 | 1/2 |

Таблиця 2.2- Розподіл ймовірностей ДВВ Y

| | | | | |
|---|---|---|-----|---|
| Y | 1 | 2 | 3 | 4 |
| q | | | 1/4 | |

Розв'язання. Скориставшись відповідним рядом розподілу ймовірностей ДВВ X та Y, знаходимо їх ентропії. Ентропія ДВВ X

$$HX = 2 \frac{1}{8} \log_2 8 + \frac{1}{4} \log_2 4 + \frac{1}{2} \log_2 2 = \frac{1}{4} 3 + \frac{1}{4} 2 + \frac{1}{2} = 1,75 \text{ (біт/сим)}.$$

Ентропія ДВВ Y $HY = \log_2 4 = 2$ (біт/сим).

Побудуємо допоміжну таблицю значень ДВВ $Z=|X-Y|$ та їх ймовірностей (табл. 2.3). Оскільки X та Y – незалежні ДВВ, то сумісна ймовірність випадання пар значень (x_i, y_j)

$$p_{ij} = P(X=i, Y=j) = p(x_i) \cdot p(y_j) = p_i \cdot q_j, \quad i, j = \overline{1, 4}.$$

Таблиця 2.3 - Сумісна ймовірність випадання пар значень (x_i, y_j)

| X | Y | | | | $\sum_j p_{ij} = p_i$ |
|-----------------------|------|------|------|------|-----------------------|
| | 1 | 2 | 3 | 4 | |
| 1 | 0 | 1 | 2 | 3 | 1/8 |
| | 1/32 | 1/32 | 1/32 | 1/32 | |
| 2 | 1 | 0 | 1 | 2 | 1/8 |
| | 1/32 | 1/32 | 1/32 | 1/32 | |
| 3 | 2 | 1 | 0 | 1 | 1/4 |
| | 1/16 | 1/16 | 1/16 | 1/16 | |
| 4 | 3 | 2 | 1 | 0 | 1/2 |
| | 1/8 | 1/8 | 1/8 | 1/8 | |
| $\sum_i p_{ij} = q_j$ | 1/4 | 1/4 | 1/4 | 1/4 | 1 |

Знайдемо ймовірності системи ДВВ $(Z=j, X=i, j = \overline{0, 3}, i = \overline{1, 4})$:

$P(Z=0, X=1)=1/32$, $P(Z=1, X=1)=1/32$, $P(Z=2, X=1)=1/32$, і т.д. (табл. 2.4).

Таблиця 2.4 – Розподіл ймовірностей системи ДВВ (X, Z)

| X | Z | | | | $\sum_j p_{ij}$ |
|-----------------|------|-------|------|------|-----------------|
| | 0 | 1 | 2 | 3 | |
| 1 | 1/32 | 1/32 | 1/32 | 1/32 | 1/8 |
| 2 | 1/32 | 1/16 | 1/32 | 0 | 1/8 |
| 3 | 1/16 | 1/8 | 1/16 | 0 | 1/4 |
| 4 | 1/8 | 1/8 | 1/8 | 1/8 | 1/2 |
| $\sum_i p_{ij}$ | 1/4 | 11/32 | 1/4 | 5/32 | 1 |

Тоді взаємна ентропія ДВВ Z та X :

$$H(Z, X) = -\sum_{ij} p_{ij} \log p_{ij} = 6 \cdot \frac{1}{32} \log 32 + 3 \cdot \frac{1}{16} \log 16 + \frac{1}{8} \log 8 = \frac{3}{16} \cdot 5 + \frac{3}{16} \cdot 4 + \frac{5}{8} \cdot 3 = \frac{57}{16} \approx 3.563 \text{ (біт/сим)}.$$

Скориставшись табл. 2.3 або табл. 2.4, побудуємо розподіл ймовірностей ДВВ Z (табл. 2.5).

Таблиця 2.5 – Розподіл ймовірностей ДВВ Z

| Z | 0 | 1 | 2 | 3 |
|----------------|-----|-------|-----|------|
| p _i | 1/4 | 11/32 | 1/4 | 5/32 |

Звідси знаходимо ентропію ДВВ Z:

$$H(Z) = \frac{2}{4} \log 4 + \frac{11}{32} \log \frac{32}{11} + \frac{5}{32} \log \frac{32}{5} = \frac{1}{2} + \frac{11}{32}(5 - \log 11) + \frac{5}{32}(5 - \log 5) = \frac{7}{2} - \frac{5}{32} \log 5 - \frac{11}{32} \log 11 \approx 1,94 \text{ біт/сим.}$$

Кількість інформації, що містить ДВВ Z стосовно ДВВ X, знаходимо, скориставшись властивістю кількості інформації і ентропії:

$$I(Z, X) = HZ + HX - H(Z, X) = \frac{7}{2} - \frac{5}{32} \log 5 - \frac{11}{32} \log 11 + \frac{7}{4} - \frac{57}{16} = \frac{27}{16} - \frac{5}{32} \log 5 - \frac{11}{32} \log 11 \approx 0,136 \text{ біт/сим.}$$

Відповідь: $H(X) = 1,75$ (біт/сим); $H(Y) = 2$ (біт/сим); $H(Z) = \frac{7}{2} - \frac{5}{32} \log 5 - \frac{11}{32} \log 11 \approx 1,94$ (біт/сим);

$$I(Z, X) = \frac{27}{16} - \frac{5}{32} \log_2 5 - \frac{11}{32} \log_2 11 \approx 0,136 \text{ (біт/сим)}; I(Z, Y) = \frac{23}{16} + \frac{3}{32} \log_2 3 - \frac{11}{32} \log_2 11 \approx 0,39 \text{ (біт/сим)}.$$

2.4 Задачі та завдання

1 Знайти ентропії ДВВ X, Y, Z і кількість інформації, що містить ДВВ $Z = (X+Y) \bmod 2$ стосовно ДВВ Y. X та Y незалежні ДВВ., задані такими розподілами:

| | | | | | | | |
|---|-----|-----|-----|-----|---|-----|-----|
| X | 0 | 1 | 3 | 4 | Y | 2 | 4 |
| P | 1/8 | 1/8 | 1/4 | 1/2 | P | 3/8 | 5/8 |

2 Скільки інформації про ДВВ X_1 містить ДВВ $Z = (X_1 + 1)^2 + X_2$, якщо незалежні ДВВ X_1 та X_2 можуть із ймовірністю $2/7$ і $5/7$ набувати значень 1 або 4? Знайти ентропії $H(X_1)$, $H(Z)$. Який характер залежності між X_1 та Z ?

3 Значення ДВВ X_1 та X_2 визначаються статистикою середньою температурою у нічний і час і вдень за останні 5 днів, а ДВВ Y є середньодобовою температурою. Знайти ентропії $H(X_1)$, $H(Y)$. Скільки інформації про ДВВ X_1 містить ДВВ Y?

4 Значення ДВВ X_1 та X_2 визначаються статистикою середньою температурою у нічний і час і вдень за останні 5 днів, а ДВВ Y є середньодобовою температурою. Знайти ентропії $H(X_1)$, $H(Y)$. Скільки інформації про ДВВ X_1 містить ДВВ Y?

3 ОСНОВИ ПОБУДОВИ КОДІВ

3.1 Класифікація кодів

Кодування інформації для зменшення надлишковості повідомень називають економним кодуванням, або стисненням інформації. Мета стиснення полягає у зменшенні кількості біт, необхідних для зберігання і передавання інформації.

Завадостійке кодування інформації за рахунок надлишковості кодів забезпечує підвищення надійності передавання даних цифровими каналами. Надлишковість коду полягає у додаванні надлишкових символів, які залежать від інформаційних. Розрізняють блокові і згорткові завадостійкі коди.

Алгоритми кодування залежать від цілей кодування, форми подання кодових слів і математичних моделей операування з кодовими словами.

3.2 Коди і алгоритми стиснення даних

Метою стиснення інформації є зменшення надлишковості подання кодових слів з врахуванням статистичних характеристик повідомлень.

Значного поширення набули алгоритми стиснення інформації на основі методів Шеннона-Фано, методів Хаффмана, блокування повідомень і використання словників груп повідомлень..

3.3 Завадостійкі коди і алгоритми їх побудови

Завадостійкість кодів визначає ціль введення надлишковості (виявлення, виправлення помилок), моделі помилок (одинарні, кратні). Алгоритми аналізу і синтезу кодів суттєво залежать від вибраного математичного апарату опису кодів (циклічні, блокові, поліноміальні тощо).

4 АЛГОРИТМИ СТИСНЕННЯ ДАНИХ

4.1 Загальна характеристика

Поширеними способами кодування є кодові таблиці, кодові дерева, метод Шеннона-Фано, метод Хаффмана. В більш складних алгоритмах застосовують блокування повідомлень, словники, чисельні подання.

4.2 Кодові таблиці і кодові дерева

Подання кодів кодовими таблицями полягає у зіставленні символам повідомлень певних кодових комбінацій фіксованої довжини, а для кодового дерева – вершин (листків) дерева

Приклад 4.1 Коди символів А-З подано у табл. 4.1, а дерево на рис. 4.1.

Таблиця 4.1 – Кодова таблиця

| Символ x_i | Число λ_i | Код з основою 10 | Код з основою 4 | Код з основою 2 |
|-----------------|----------------------|------------------|-----------------|-----------------|
| <i>А</i> | 0 | 0 | 00 | 000 |
| <i>Б</i> | 1 | 1 | 01 | 001 |
| <i>В</i> | 2 | 2 | 02 | 010 |
| <i>Г</i> | 3 | 3 | 03 | 011 |
| <i>Д</i> | 4 | 4 | 10 | 100 |
| <i>Е</i> | 5 | 5 | 11 | 101 |
| <i>Ж</i> | 6 | 6 | 12 | 110 |
| <i>З</i> | 7 | 7 | 13 | 111 |

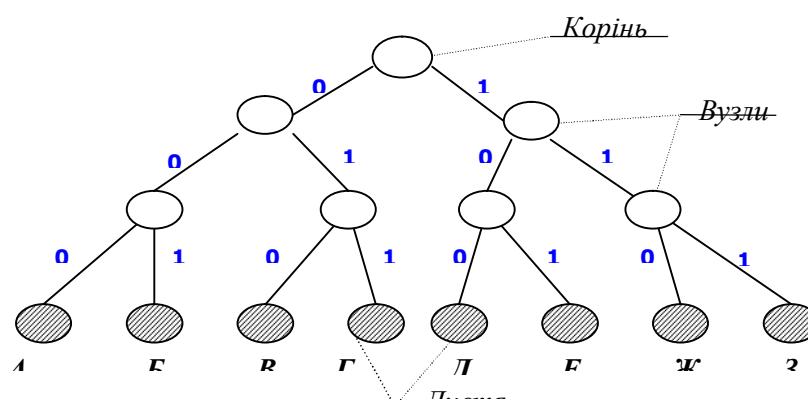


Рисунок 4.1 – Кодове дерево

Алгоритм побудови бінарного дерева починає з деякої точки - кореня кодового дерева. З кореня проводять гілки, що позначаються 0 або 1 (для двійкового подання), з яких також формують гілки до досягнення рівня листків. Листям кодового дерева відповідають символи алфавіту джерела, а шлях від кореня до відповідного листа визначає певну кодову комбінацію (кодове слово). Сукупність всіх кодових слів (кодів символів) утворює код джерела.

4.3 Алгоритм кодування за методом Шеннона – Фано

У методі Шеннона-Фано значення ДВВ символів розміщаються у порядку спадання ймовірностей. Далі уся сукупність розділяється на дві приблизно рівні за сумою ймовірностей частини. До коду першої частини додається 0, а до коду другої - 1. Кожна з частин за тим самим принципом знову розділяється (якщо це можливо) на дві частини і т.і.

Приклад 4.2 Побудова таблиці кодів символів за методом Шеннона-Фано для повідомень, заданих розподілом ймовірностей, подано у табл.. 4.2.

Таблиця 4.2 – Кодування символів за методом Шеннона -Фано

| Буква, x_i | Імовірність, p_i | Код | Довжина коду, l_i | $p_i l_i$ |
|----------------|--------------------|------------------------|---------------------|-----------|
| А | 0,6 | 1} | 1 | 0,6 |
| Б | 0,2 | 0 1} | 2 | 0,4 |
| В | 0,1 | 0 0 1} | 3 | 0,3 |
| Г | 0,04 | 0 0 0 1} | 4 | 0,16 |
| Д | 0,025 | 0 0 0 0 1} | 5 | 0,125 |
| Е | 0,015 | 0 0 0 0 0 1} | 6 | 0,09 |
| Ж | 0,01 | 0 0 0 0 0 0 1} | 7 | 0,07 |
| З | 0,01 | 0 0 0 0 0 0 0 1} | 7 | 0,07 |
| $\sum p_i = 1$ | | $\sum p_i l_i = 1,815$ | | |

$$\bar{L} = \sum_{i=1}^8 l_i p_i = 1,815 \quad (\text{біт/сим}).$$

Середня довжина коду

$$\rho_k = 1 - \frac{H(X)}{\bar{L}} = 1 - \frac{1,781}{1,825} \approx 0,019$$

Надлишковість коду , тобто на порядок менше, ніж для рівномірного кодування.

4.4 Алгоритм кодування за методом Хаффмана

Алгоритм створення кода полягає у побудові бінарного дерева. Ймовірності ДВВ розміщуються у порядку спадання і приписують листам кодового дерева. Величину, що приписують вузлу дерева, називають його вагою. Два листи або вузли з найменшими значеннями ваги утворюють батьківський вузол, вага якого дорівнює сумарній вазі вузлів, що його складають. Надалі цей вузол враховується нарівні з вершинами, що залишилися, а листя або вузли, що його утворили, більше не розглядаються.

Приклад 4.3 Кодове дерево наведено на рис. 4.2, а відповідна таблиця кодів в табл.4.3.

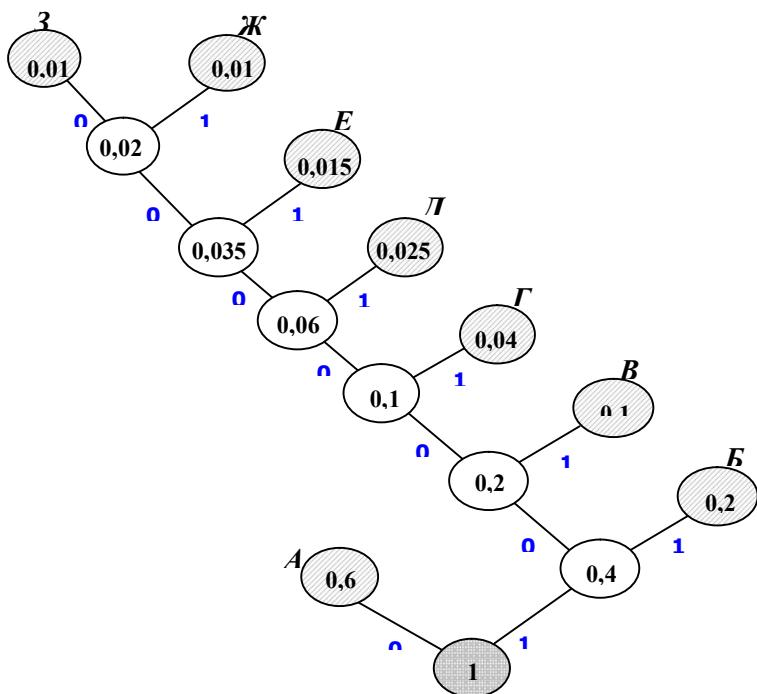


Рисунок 4.2 – Код Хаффмана

Після побудови кореня кожна визначена гілка, що виходить з батьківського вузла, позначається 0 (зазвичай це ліва гілка) або 1 (права гілка). Коди значень ДВВ – це послідовності 0 і 1, що утворюються, на шляху від кореня кодового дерева до листа із заданою імовірністю ДВВ.

Таблиця 4.3 – Код Хаффмана

| Буква, x_i | Ймовірність, p_i | Код, Code(X_i) | Довжина коду, l_i | $l_i p_i$ |
|----------------|--------------------|--------------------|---------------------|------------------------|
| А | 0,6 | 0 | 1 | 0,6 |
| Б | 0,2 | 1 1 | 2 | 0,4 |
| В | 0,1 | 1 0 1 | 3 | 0,3 |
| Г | 0,04 | 1 0 0 1 | 4 | 0,16 |
| Д | 0,025 | 1 0 0 0 1 | 5 | 0,125 |
| Е | 0,015 | 1 0 0 0 0 1 | 6 | 0,09 |
| Ж | 0,01 | 1 0 0 0 0 0 1 | 7 | 0,07 |
| З | 0,01 | 1 0 0 0 0 0 0 | 7 | 0,07 |
| $\sum p_i = 1$ | | | | $\sum p_i l_i = 1,815$ |

Середня довжина коду $\bar{L} = \sum_{i=1}^k l_i p_i = 1,815$ (біт/сим). Надлишковість коду $\rho_\kappa = 1 - \frac{1,781}{1,815} \approx 0,019$.

4.5 Алгоритм блокового кодування за методом Хаффмана

Блоковий код розділяє вектор даних на блоки певної довжини (порядок коду), і потім кожний блок замінюють кодовим словом з префіксної множини кодових слів. Отриману послідовність кодових слів об'єднують в остаточну двійкову послідовність на виході кодера.

За Шенноном верхня границя стиснення інформації є ентропією джерела $H(X)$. Середнє значення довжини коду обчислюють як $\overline{L(X)} = \sum_{i=1}^k p(x_i) L(X_i)$,

$L(X) = \text{len}(\text{code}(X_i))$, де $\text{code}(X_i)$ значенню X_i ставить у відповідність деякий бітовий код; $\text{len}()$ - повертає довжину цього коду.

Наслідком теореми Шеннона про кодування джерела у відсутності шуму

$$\epsilon \overline{L(X)} \geq HX \text{ для будь-якої ДВВ } X \text{ і будь-якого її коду.}$$

Блоковий код називається блоковим кодом k -го порядку, якщо всі його блоки мають довжину k символів. За заданим $\epsilon > 0$ існує така довжина блоку k (для повідомлення довжиною n буде n/k блоків), що за умови використання оптимального статистичного кодування блоків можна досягти середньої довжини коду більше ентропії менш ніж на ϵ .

Дійсно, для незалежних ДВВ X_1, X_2, \dots, X_n з однаковим розподілом

$$\text{ймовірностей ентропія } n\text{-вимірної ДВВ } \vec{X} = (X_1, X_2, \dots, X_n) \quad \overrightarrow{HX} = n\overrightarrow{HX}_1.$$

Для блокового повідомлення джерела $\vec{Y} = (\vec{Y}_1, \vec{Y}_2, \dots, \vec{Y}_{n/k})$

$$\vec{Y}_1 = (X_1, X_2, \dots, X_k), \quad \vec{Y}_2 = (X_{k+1}, X_{k+2}, \dots, X_{2k}), \quad \vec{Y}_i = (X_{k(i-1)+1}, X_{k(i-1)+2}, \dots, X_{ki}) \text{ - блоки}$$

$$\text{повідомлення } \overrightarrow{HY}_1 = k\overrightarrow{HX}_1.$$

При оптимальному кодуванні k -послідовностей векторної ДВВ. \vec{Y} , що

$$\text{розглядаються як одиниці повідомлення } \overline{L(\vec{Y}_1)} \leq \overrightarrow{HY}_1 + 1.$$

Середня кількість біт на одиницю повідомлення X $\overline{L(X)} = \frac{\overline{L(\vec{Y}_i)}}{k}$. Звідси

$$k\overline{L(X)} \leq k\overrightarrow{HX} + 1 \text{ і } \overline{L(X)} \leq \overrightarrow{HX} + \frac{1}{k}, \text{ тобто достатнім є вибір } k = 1/\epsilon.$$

Приклад 4.4 ДВВ X задана розподілом ймовірностей: $P(X=A)=1/3$; $P(X=B)=7/15$; $P(X=C)=1/5$. Побудувати таблицю кодів для блокового коду Хаффмана другого порядку. Визначити середню довжину коду.

Для побудови коду будуємо розподіл ймовірностей векторної ДВВ $\vec{X} = (X_1, X_2)$, що являє собою блок повідомлення довжиною в два символи (таблиця 4.4).

Виходячи з отриманого ряду ймовірностей, побудуємо кодове дерево за алгоритмом Хаффмана для значень ДВВ \vec{X} (рис.4.3) і відповідну таблицю 4.4 кодових слів.

Таблиця 4.4 – Таблиця кодових слів блокового коду

| \vec{X} | BB | BA | AB | AA | BC | CB | AC | CA | CC |
|-----------------|-----------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| $P(\vec{X})$ | 49/225 | 7/45 | 7/45 | 1/9 | 7/75 | 7/75 | 1/15 | 1/15 | 1/25 |
| $Code(\vec{X})$ | 10 | 001 | 010 | 011 | 111 | 0000 | 0001 | 1100 | 1101 |
| $L(\vec{X})$ | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 |
| $P_i \cdot L_i$ | 98/225 | 7/15 | 7/15 | 1/3 | 7/25 | 28/75 | 4/15 | 4/15 | 4/25 |

Середня довжина коду для блокового коду Хаффмана 2-го порядку

$$\overline{L(X)} = \frac{\sum P_i \cdot L_i}{n} = \frac{686}{225} / 2 = \frac{686}{450} \approx 1,524 \text{ (біт/сим)}.$$

Для порівняння: наведемо кодове дерево (рис.4.4) і відповідну таблицю кодів (таблиця 4.5) для одновимірної ДВВ

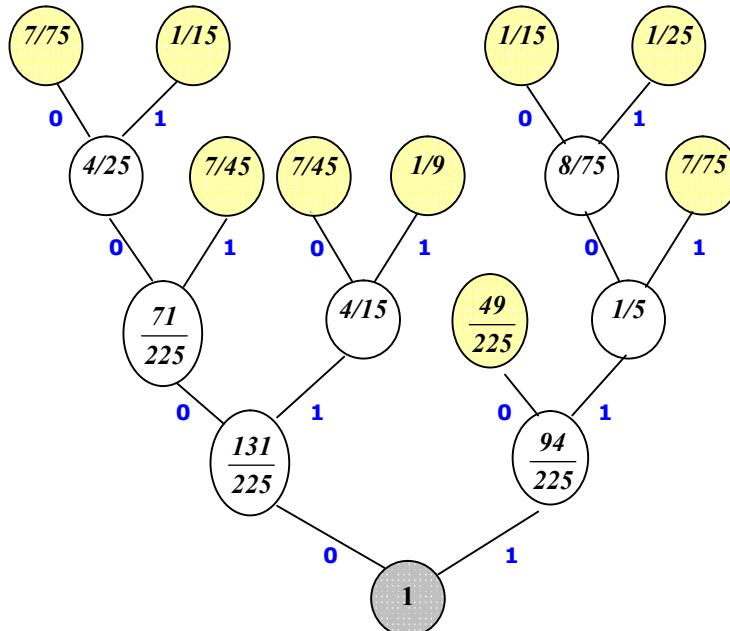


Рисунок 4.3 – Кодове дерево

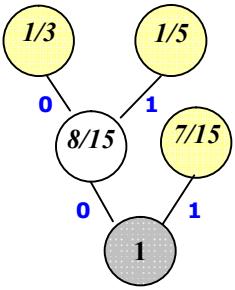


Рисунок 4.4 – Кодове дерево

Таблиця 4.5 – Таблиця кодування символів

| x_i | B | A | C | Σ |
|-------------|------|-----|-----|----------|
| p_i | 7/15 | 1/3 | 1/5 | 1 |
| $Code(x_i)$ | 1 | 00 | 01 | |
| l_i | 1 | 2 | 2 | |
| $p_i l_i$ | 7/15 | 2/3 | 2/5 | 23/15 |

$$\overline{L(X)} = \frac{23}{15} \approx 1,533 \text{ (біт/сим)}.$$

Мінімальна середня довжина коду для кодування даної ДВВ визначається її ентропією:

$$\overline{L}_{\min} = HX = \frac{7}{15} \log_2 \frac{15}{7} + \frac{1}{3} \log_2 3 + \frac{1}{5} \log_2 5 = 1,506 \text{ (біт.симв)}$$

Отже, надлишковість блокового коду $\rho_\kappa = 1 - \frac{1,506}{1,524} \approx 0,012$, неблокового

$\rho_\kappa = 1 - \frac{1,506}{1,533} \approx 0,017$, а рівномірного коду $\rho_\kappa = 1 - \frac{1,506}{2} \approx 0,25$, що істотно більше.

Приклад 4.5 Закодувати повідомлення АВАААВВА за алгоритмом Хаффмана і за блоковим алгоритмом Хаффмана 2-го порядку. Обчислити довжини отриманих кодів. Приблизний закон розподілу ймовірностей визначити з аналізу повідомлення.

Побудуємо приблизний закон розподілу їх ймовірностей: $P(A)=5/8$, $P(B)=3/8$ з кодами 0 і 1 за алгоритмом Хаффмана. За цими кодами кодуємо повідомлення: $Code(\text{ABAABBA})=01000110$. Довжина коду $L(X)=8$ біт.

Блокове кодування повідомлення АВАААВВА на 4 блоки довжиною 2 символи дозволяє отримати приблизний статистичний закон розподілу

ймовірностей блоків: $P(AB) = 1/2$, $P(AA)=1/4$, $P(BA)=1/4$, а за алгоритмом Хаффмана відповідні коди 0, 10, 11. Блоковий код повідомлення Code(ABAABBA)=010011 має довжина $L(X)=6$ біт.

4.6 Алгоритм арифметичного кодування

Спочатку за розподілом ймовірностей ДВВ складається таблиця з пересічних в граничних точках відрізків для кожного із значень ДВВ. Об'єднання цих відрізків утворює інтервал $[0;1]$, а їхні довжини пропорційні ймовірностям значень ДВВ.

Алгоритм кодування полягає в побудові інтервалу, що однозначно визначає конкретну послідовність значень ДВВ у такий спосіб. З відрізу повідомлення завдовжки $n-1$ символів для побудови відрізка повідомлення завдовжки n попередній інтервал розбивається на стільки частин, скільки можливих значень має ДВВ. Для знаходження початку і кінця нового інтервалу повідомлення до початку попереднього інтервалу необхідно додати значення добутків його ширини на відповідні граници відрізка поточного нового символу з таблиці символів і їхніх інтервалів (таблиці кодера). З отриманих інтервалів вибирається той, що відповідає конкретному повідомленню завдовжки n символів.

Для побудованого таким чином інтервалу повідомлення знаходиться число, що належить цьому відрізу. Зазвичай це ціле число, розділене на мінімальний степінь 2. Це дійсне число і буде кодом даного повідомлення.

За надходженням символів повідомлення його інтервал звужується, а відповідна кількість розрядів, необхідна для подання інтервалу, збільшується. Більш імовірні символи меншою мірою звужують інтервал, ніж менш імовірні, і, отже, додають менше розрядів до результату.

Приклад 4.6 Побудувати код слова «МАТЕМАТИКА».

Частоти символів і відрізки символів слова «МАТЕМАТИКА» подані таблиці 4.6.

Таблиця 4.6 – Імовірності появи і інтервали символів

| Символ | Імовірність | Інтервал |
|--------|-------------|------------|
| M | 0,2 | [0; 0,2) |
| A | 0,3 | [0,2; 0,5) |
| T | 0,2 | [0,5; 0,7) |
| E | 0,1 | [0,7; 0,8) |
| И | 0,1 | [0,8; 0,9) |
| K | 0,1 | [0,9; 1,0) |

Для першого символу «M» кодер звужує початковий інтервал [0; 1) до нового [0; 0,2). Наступний символ «A» кодується підінтервалом [0,04;0,1): lowi=0+0,2*0,2=0,04; highi=0+0,2*0,5=0,1).

Букві «T» відповідає інтервал [0,5; 0,7), а новий інтервал буде [0,07; 0,082) (lowi=0,04+0,06*0,5=0,07; highi=0,04+0,06*0,7=0,082).

Послідовність інтервалів кодування повідомлення «МАТЕМАТИКА» наведена в таблиці 4.7.

Таблиця 4.7 – Кодування повідомлення

| Символ – інтервал | Інтервал повідомлення | Ширина інтервалу |
|-------------------|-----------------------------|------------------------|
| M - [0; 0,2) | [0; 0,2) | 0,2 |
| A - [0,2; 0,5) | [0,04; 0,1) | 0,06 |
| T - [0,5; 0,7) | [0,07; 0,082) | 0,012 |
| E - [0,7; 0,8) | [0,0784; 0,0796) | 0,00012 |
| M - [0; 0,2) | [0,0784; 0,07864) | 0,00024 |
| A - [0,2; 0,5) | [0,078448; 0,07852) | 0,72·10 ⁻⁴ |
| T - [0,5; 0,7) | [0,078484; 0,0784984) | 0,144·10 ⁻⁴ |
| И - [0,8; 0,9) | [0,07849552; 0,07849696) | 0,144·10 ⁻⁵ |
| K - [0,9; 1,0) | [0,078496816; 0,07849696) | 0,144·10 ⁻⁶ |
| A - [0,2; 0,5) | [0,0784968448; 0,078496888) | 0,432·10 ⁻⁷ |

З результату кодування повідомлення «МАТЕМАТИКА» отримаємо інтервал [0,0784968448; 0,078496888). Для цього числа знаходять мінімальний степінь 2 - це $0,07849687 = \frac{1316959}{2^{24}}$, яка визначає 24 - розрядний код повідомлення. Двійковий 24-розрядний код числа 131625910 = =0001010000011000010111112 є арифметичним кодом даного повідомлення: Code(МАТЕМАТИКА)=00010100000110000101111.

Довжина коду $L(X)=24$ біт а середня довжина $\overline{L(X)} = 24/10 = 2,4$ (біт/сим).

Арифметичний код можна будувати для повідомлень алфавіту.

Приклад 4.7 ДВВ може набувати значень 0 і 1 з ймовірностями відповідно $2/3$ і $1/3$. Побудуємо арифметичний код для повідомлень завдовжки 3 символи.

Коди за арифметичним алгоритмом для повідомлень завдовжки 3 символи для заданої ДВВ подаються таблицею 4.8.

Таблиця 4.8 – Арифметичне кодування послідовності символів

| Повідомлення та їх інтервали | | | | | Код | p_i | |
|------------------------------|------------|----|--------------|-----|--------------------------------------|-------|--------|
| 1 | $[2/3; 1)$ | 11 | $[8/9; 1)$ | 111 | $[26/27; 1] \ni 31/32 \rightarrow$ | 1111 | $1/27$ |
| | | | | 110 | $[8/9; 26/27] \ni 15/16 \rightarrow$ | 111 | $2/27$ |
| | | 10 | $[2/3; 8/9)$ | 101 | $[22/27; 8/9] \ni 7/8 \rightarrow$ | 111 | $2/27$ |
| | | | | 100 | $[2/3; 22/27] \ni 3/4 \rightarrow$ | 11 | $4/27$ |
| 0 | $[0; 2/3)$ | 01 | $[4/9; 2/3)$ | 101 | $[16/27; 2/3] \ni 5/8 \rightarrow$ | 101 | $2/27$ |
| | | | | 100 | $[4/9; 16/27] \ni 1/2 \rightarrow$ | 1 | $4/27$ |
| | | 00 | $[0; 4/9]$ | 001 | $[8/27; 4/9] \ni 3/8 \rightarrow$ | 011 | $4/27$ |
| | | | | 000 | $[0; 8/27] \ni 1/4 \rightarrow$ | 01 | $8/27$ |

Середня

довжина

коду

$$\overline{L(X)} = \frac{1}{n} \sum_{i=1}^k L(\vec{X}_i) \cdot p_i =$$

$$= \frac{1}{81} (5 \cdot 1 + 4 \cdot 2 + 3 \cdot 2 + 2 \cdot 4 + 3 \cdot 2 + 1 \cdot 4 + 3 \cdot 4 + 2 \cdot 8) = \frac{65}{81} \approx 0,8025 \text{ (біт/сим)}.$$

Декодування арифметичного коду повідомлення здійснюється за таким алгоритмом. Декодеру і кодеру відома таблиця розподілу інтервалів символів алфавіту.

Крок 1 За таблицею інтервалів символів алфавіту визначають відрізок, що містить значення поточного коду, і за ним однозначно визначають символ повідомлення. Якщо символ - маркер кінця повідомлення, то кінець, інакше - пейти до кроку 2.

Крок 2 Від поточного коду віднімається нижня границя його інтервалу. Різниця ділиться на довжину цього інтервалу. Отримане значення вважається новим значенням поточного коду. Перейти до кроку 1.

Приклад 4.8 Арифметичний код повідомлення з 10 символів є 0001010000011000010111112 = 131625910, а символи і інтервали повідомлення наведені у таблиці 4.8. Декодувати це повідомлення.

Дійсне число, яке належить інтервалу, що однозначно визначає

закодоване повідомлення $\frac{1316959}{2^{24}} = 0,07849687$. Це число є значенням поточного коду. З таблиці 4.8 визначають відрізок [0; 0,2), якому належить це число і перший закодований символ «М». Віднімаємо від поточного коду нижню границю інтервалу символу «М» і ділимо отриманий результат на

ширину цього інтервалу: $\frac{0,07849687 - 0}{0,2} = 0,39248435$. Це число належить відрізку [0,2; 0,5), що відповідає символу «А». Виключимо з отриманого інтервалу [0,2; 0,5) вплив букви «А» відніманням нижньої границі цього

інтервалу і розділимо на його ширину: $\frac{0,39248435 - 0,2}{0,3} = 0,6416145$. Результат належить відрізку [0,5; 0,7) символу «Т» і т.і. Отримаємо таке повідомлення (Табл. 4.9).

Таблиця 4.9 – Декодування повідомлення

| Декодоване число | Символ на виході | Інтервал | Ширина інтервалу |
|------------------|------------------|------------|------------------|
| 0,07849687 | M | [0; 0,2) | 0,2 |
| 0,39248435 | A | [0,2; 0,5) | 0,3 |
| 0,6416145 | T | [0,5; 0,7) | 0,2 |
| 0,7080725 | E | [0,7; 0,8) | 0,1 |
| 0,080725 | M | [0; 0,2) | 0,2 |
| 0,403625 | A | [0,2; 0,5) | 0,3 |
| 0,67875 | T | [0,5; 0,7) | 0,2 |
| 0,89375 | I | [0,8; 0,9) | 0,1 |
| 0,9375 | K | [0,9; 1,0) | 0,1 |
| 0,375 | A | [0,2; 0,5) | 0,3 |

4.7 Задачі та завдання з кодів Шеннона-Фано і Хаффмана

- 1 Побудувати таблиці кодів і обчислити їх середню довжину за алгоритмами Шеннона-Фано, Хаффмана і арифметичним для дискретної випадкової величини X , заданої таким розподілом:

| | | | | | | | |
|-----|------|-----|-----|-----|------|-----|-----|
| X | 1 | 4 | 9 | 16 | 25 | 36 | 49 |
| P | 0,05 | 0,1 | 0,1 | 0,3 | 0,15 | 0,1 | 0,2 |

- 2 Обчислити середню довжину кодів Шеннона-Фано, Хаффмана і арифметичного кода для значень дискретної випадкової величини X , заданої таким розподілом ймовірностей:

| | | | | | |
|-----|-----|-----|-----|-----|------|
| X | -2 | -1 | 0 | 1 | 2 |
| P | 1/3 | 1/4 | 1/5 | 1/6 | 1/20 |

- 3 Обчислити середню довжину кодів Шеннона-Фано, Хаффмана і арифметичного кода для значень дискретної випадкової величини X , заданої таким розподілом ймовірностей:

| | | | | | | | |
|-----|-----|-----|-----|------|-----|------|-----|
| X | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| P | 0,1 | 0,2 | 0,1 | 0,05 | 0,1 | 0,05 | 0,4 |

4.8 Адаптивний алгоритм Хаффмана з упорядкованим деревом

В адаптивному (динамічному) алгоритмі Хаффмана таблиця кодів не передається. У процесі кодування залежно від значення поточного символу, що надходять на вхід алгоритму, кодове дерево коригується відповідно до зміни статистики вхідного потоку. При декодуванні відбувається той самий процес. Для однозначності декодування використовується упорядкована структура кодового дерева.

Упорядкованим деревом Хаффмана називається бінарне дерево, вузли якого можуть бути перелічені у порядку неубування їх ваги зліва-направо на кожному рівні і знизу-вверх за рівнями. При зміні ваги існуючого вузла в

дереві достатньо поміняти місцями два вузли: вузол, що порушив упорядкованість, і останній з наступних за ним вузлів меншої ваги. Після обміну вузлів місцями необхідно перерахувати вагу всіх вузлів-предків.

На початку роботи алгоритму дерево містить тільки один спеціальний символ <ESC>, що завжди має частоту 0. Він необхідний для занесення в кодове дерево нового символу, що передається безпосередньо після <ESC>. При появі нового символу праворуч від вузла <ESC> додається лист і потім, якщо необхідно, дерево упорядковується. Ліві гілки кодового дерева позначаються 0, а праві - 1.

Так при додаванні у дерево Хаффмана рис. 6.1а двох букв необхідно поміняти місцями вузли «A» і «D» (рис. 4.5б). При додаванні двох нових букв «A» потрібно поміняти місцями спочатку вузол «A» та вузол - батько «D» і «B», а потім вузол «E» і вузол - брат «E» (рис. 4.5 в-г).

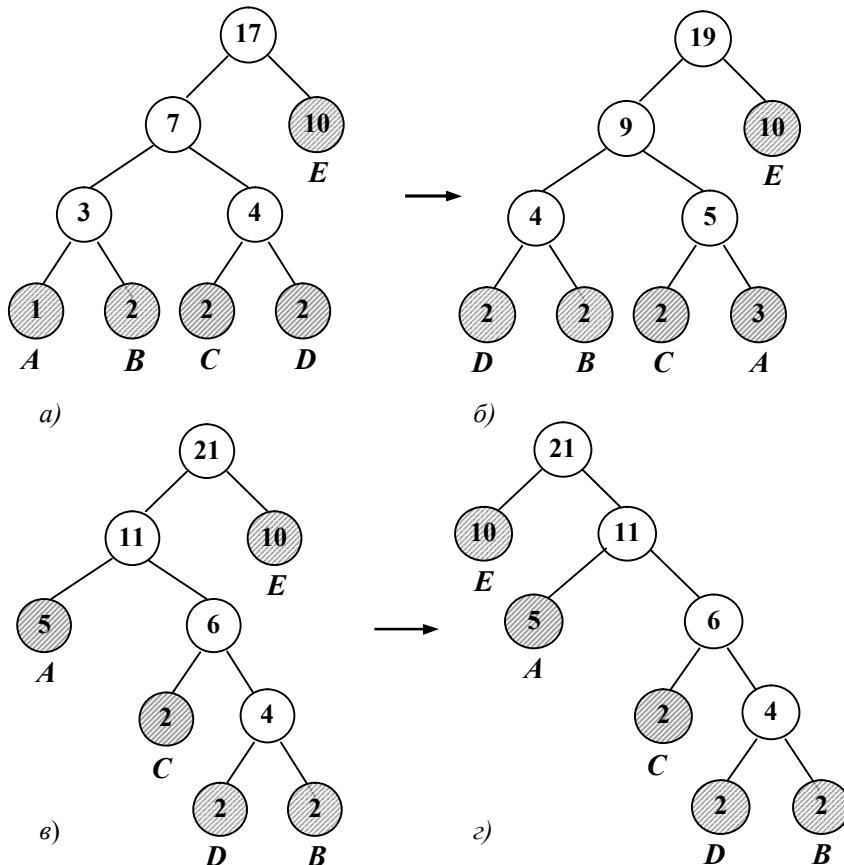


Рисунок 4.5 – Коригування дерева Хаффмана

Приклад 4.9 Закодувати повідомлення СИНЯЯ СИНЕВА СИНИ за адаптивним алгоритмом Хаффмана. Обчислити довжини коду.

Процес кодування повідомлення і відповідні зміни кодового дерева подаються в таблиці 4.10 та рис. 4.6.

Таблиця 4.10 – Опис потоку вхідних символів

| Вхідні дані | Код | Довжина Коду | Номер дерева |
|-------------|-----------|--------------|--------------|
| C | 'C' | 8 | 1 |
| I | '0'I' | 9 | 2 |
| H | '00'H' | 10 | 3 |
| Я | '100'Я' | 11 | 4 |
| Я | '001' | 3 | 5 |
| | '100' ' | 11 | 6 |
| C | '101' | 3 | 7 |
| I | '00' | 2 | 8 |
| H | '101' | 3 | 9 |
| E | '1100'E' | 12 | 10 |
| B | '11000'B' | 13 | 11 |
| A | '10100'A' | 13 | 12 |
| | '1010' | 4 | 13 |
| C | '101' | 3 | 14 |
| I | '101' | 3 | 15 |
| H | '101' | 3 | 16 |
| I | '111' | 3 | |

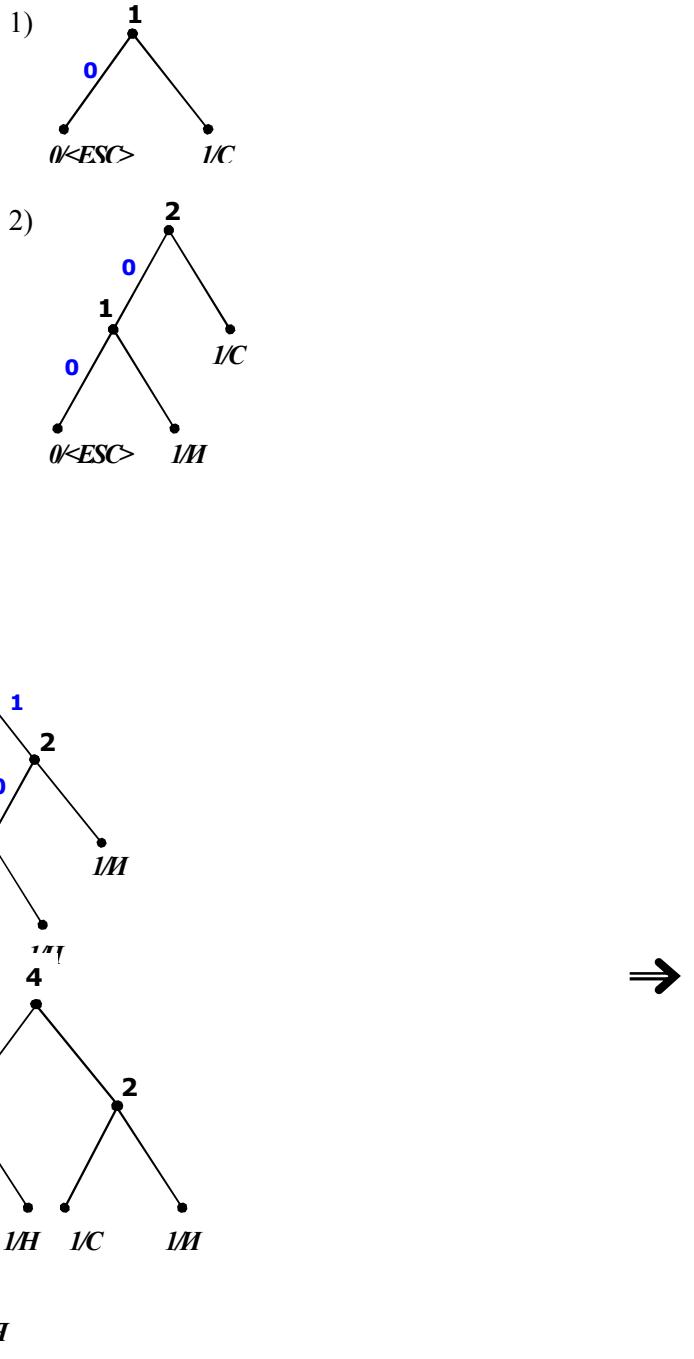


Рисунок 4.6–Дерева Хаффмана

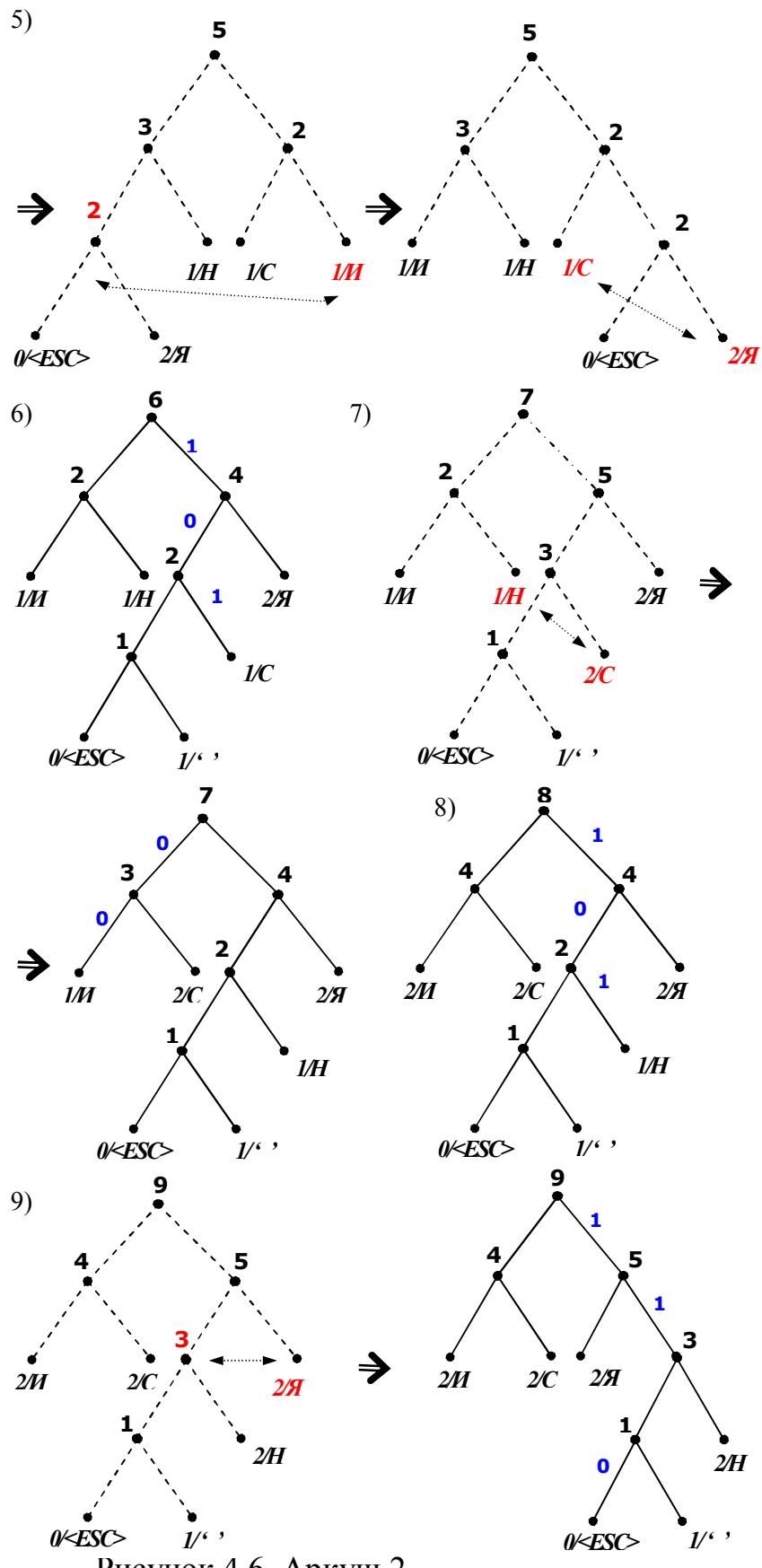


Рисунок 4.6–Аркуш 2

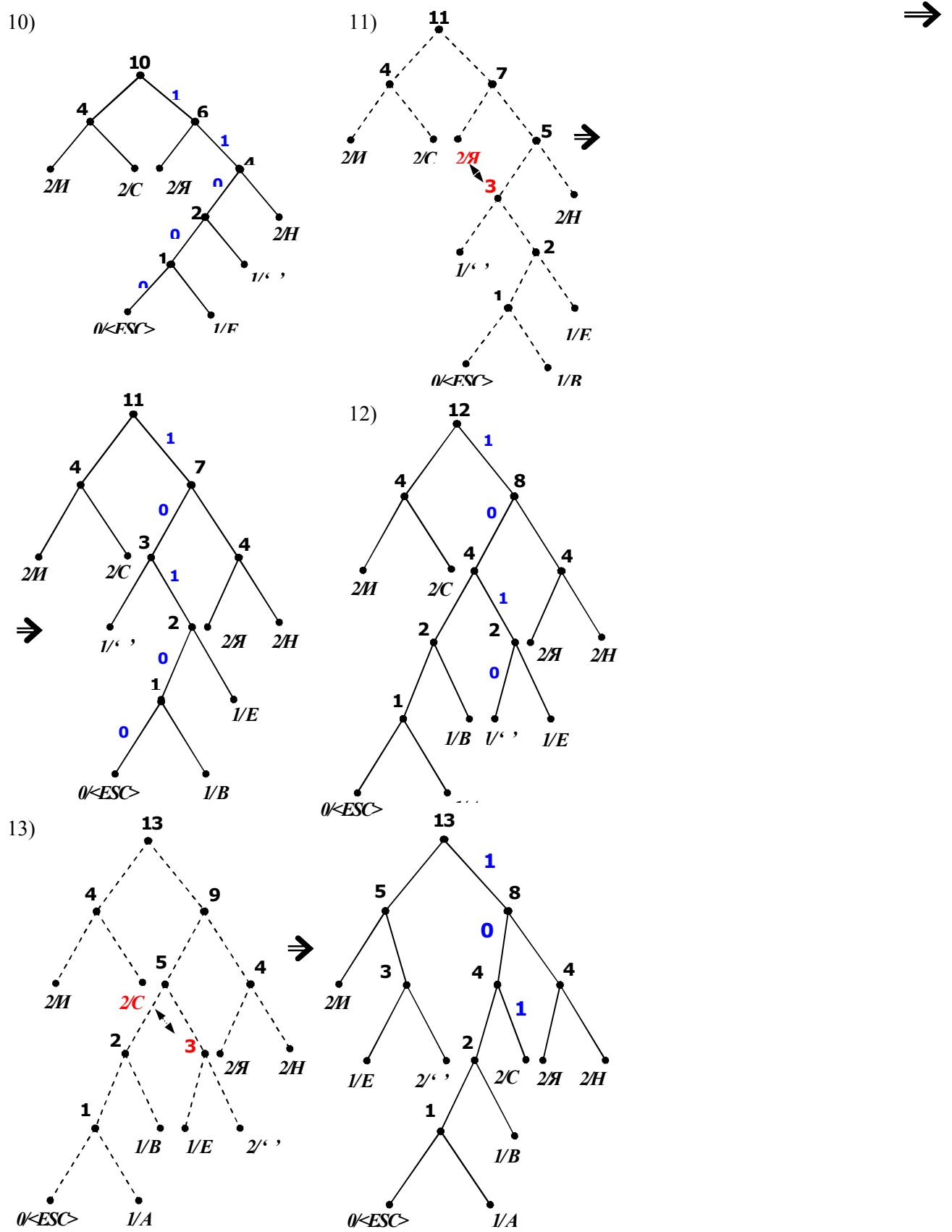
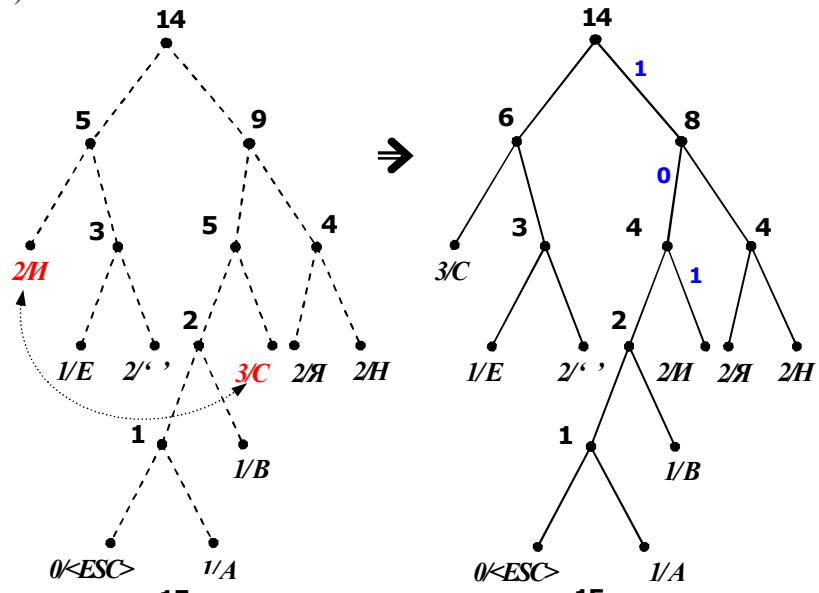
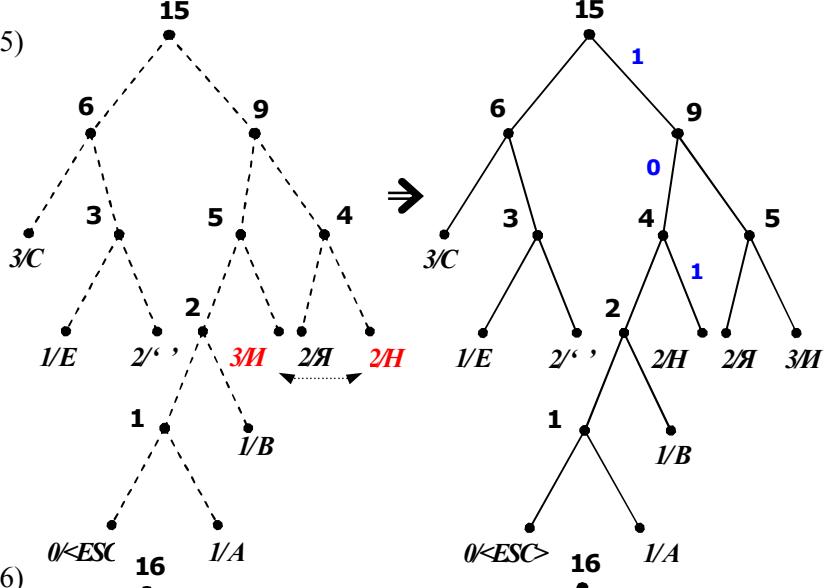


Рисунок 4.6–Аркуш 3

14)



15)



16)

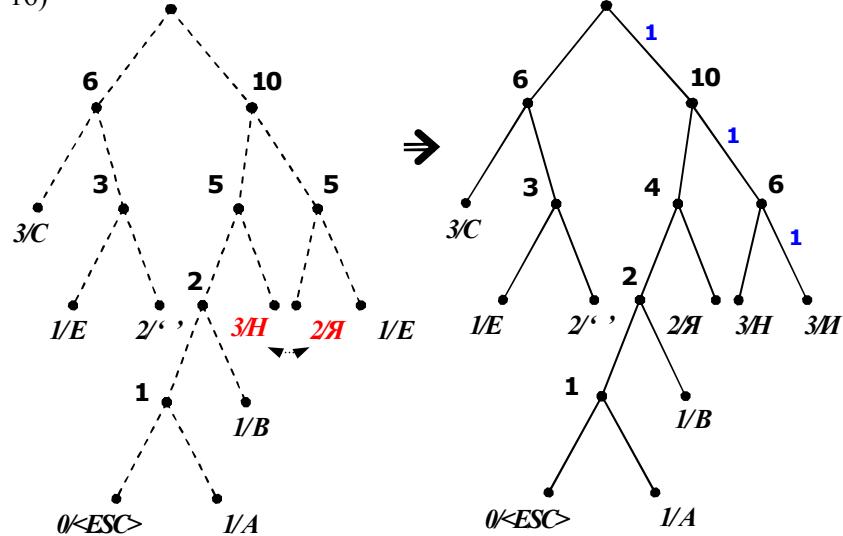


Рисунок 4.6–Аркуш 4

Довжина коду стиснутого повідомлення $L(X)=114$ біт.

Довжина ASCII+коду нестиснутого повідомлення $L(X)=136$ біт.

Приклад 4.10 Декодувати повідомлення 'B'0'D'00'C'11111
110101011011110100101, закодоване за адаптивним алгоритмом Хаффмана.
Обчислити довжину стиснутого і нестиснутого повідомлення у бітах.

Процес декодування подано у таблиці 4.11, а побудова кодового дерева на рис. 4.7.

Таблиця 4.11 – Опис декодування символів за повідомленням

| Вхідний код | Символ | Довжина Коду | Номер дерева |
|-------------|--------|--------------|--------------|
| 'B' | B | 8 | 1 |
| 0'D' | D | 9 | 2 |
| 00'C' | C | 10 | 3 |
| 11 | D | 2 | 4 |
| 11 | B | 2 | 5 |
| 11 | B | 2 | 6 |
| 101 | C | 3 | 7 |
| 0 | B | 1 | 8 |
| 101 | C | 3 | 9 |
| 101 | D | 3 | 10 |
| 11 | C | 2 | 11 |
| 101 | D | 3 | 12 |
| 0 | B | 1 | 13 |
| 0 | B | 1 | 14 |
| 101 | D | 3 | |
| $\Sigma=53$ | | | |

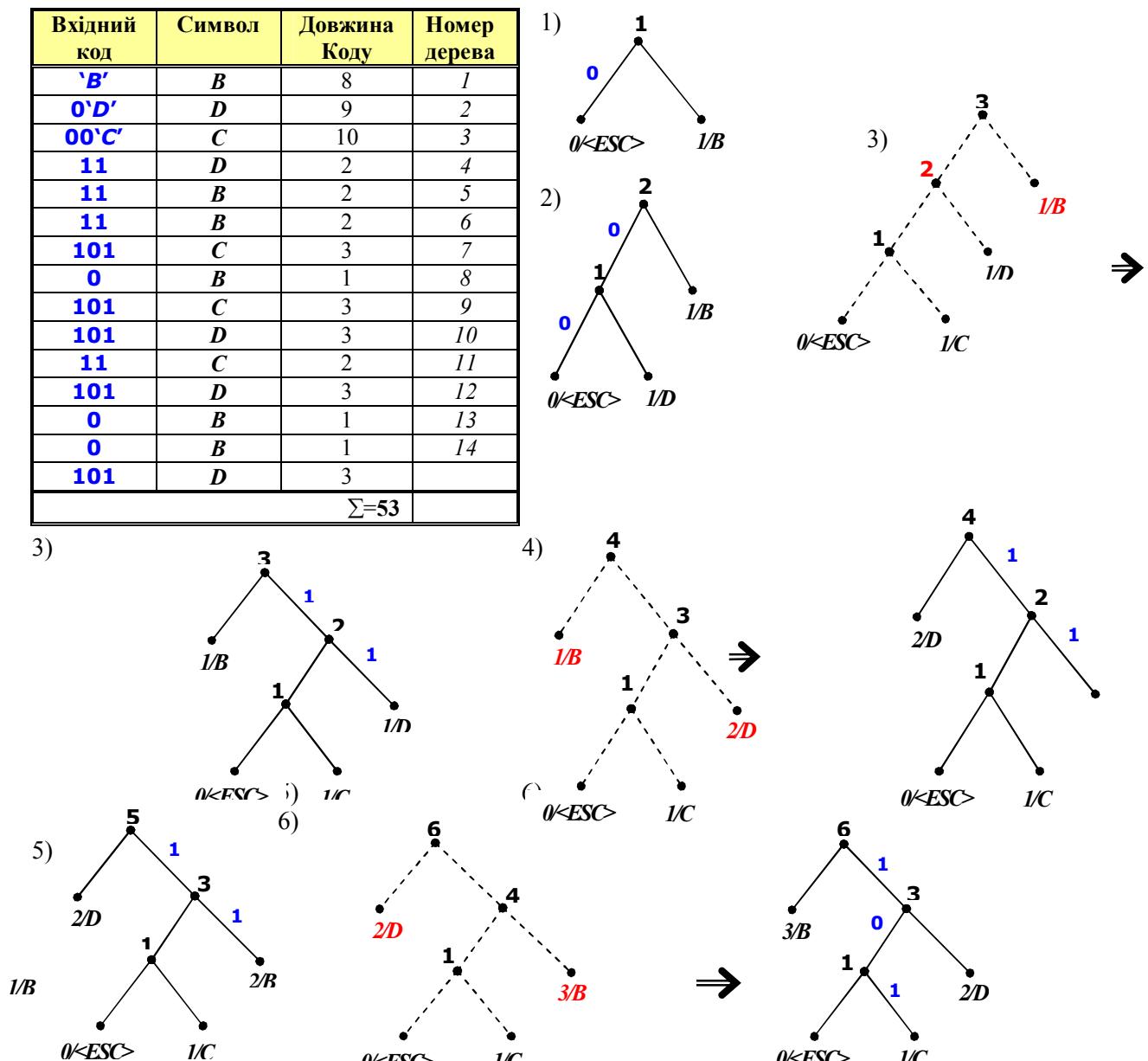


Рисунок 4.7 – Декодування повідомлення

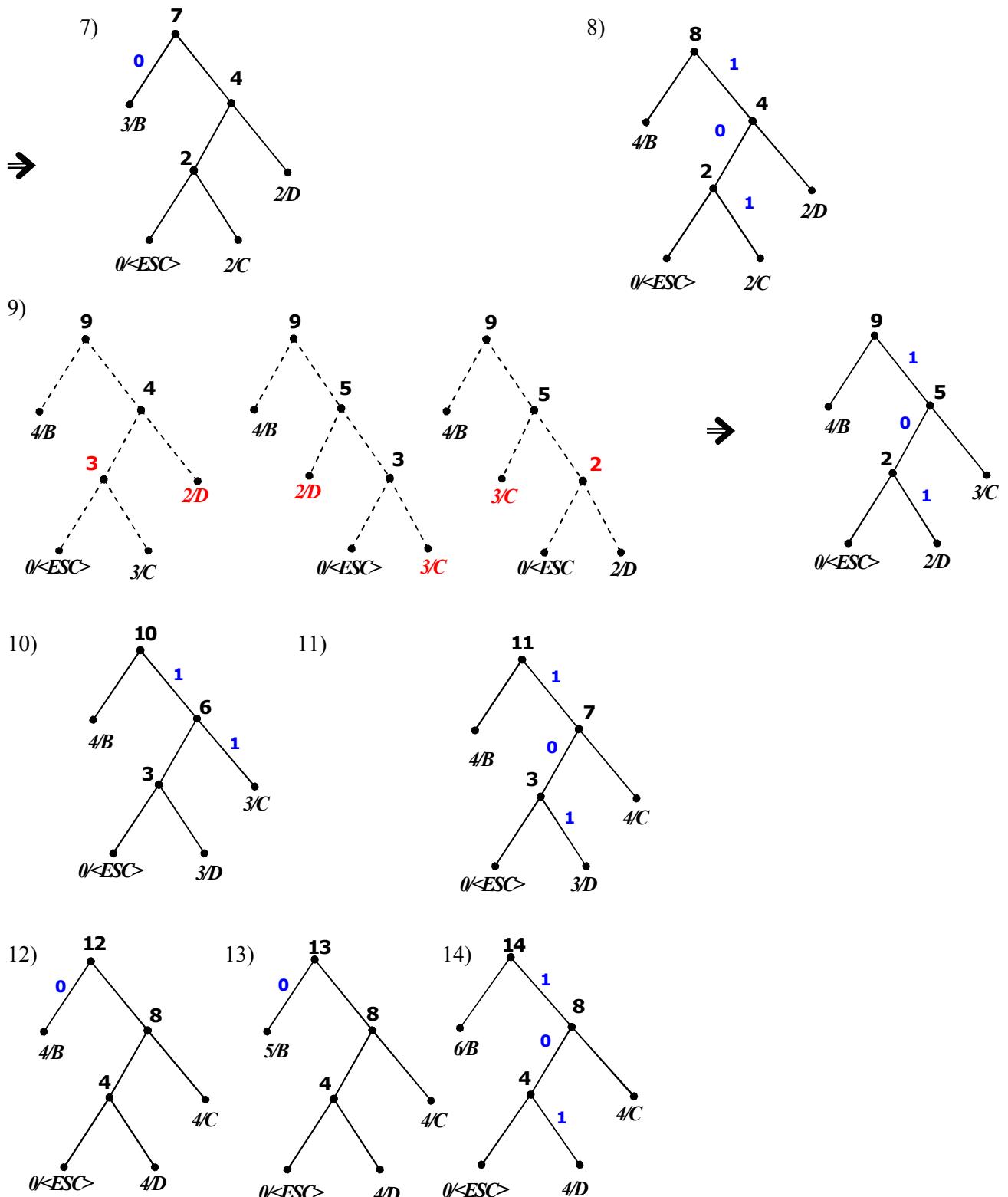


Рисунок 4.7 – Аркуш 2

Отже, закодованим є повідомлення BDCDBBCBCDCDBBD. Довжина стиснутого повідомлення $L(X)=53$ біт. Довжина нестиснутого повідомлення в коді ASCII+ $L(X)=15 \cdot 8=120$ біт.

4.9 Задачі та завдання за адаптивним алгоритмом Хаффмана

- 1 Закодувати повідомлення AABDBBCDAAACCCDDBB за адаптивним алгоритмом Хаффмана. Обчислити довжину у бітах стиснутого повідомлення і його ASCII+-коду.
- 2 Закодувати повідомлення EMBEDDED за адаптивним алгоритмом Хаффмана. Обчислити довжини в бітах стиснутого повідомлення і його ASCII+-коду.
- 3 Розкодувати повідомлення ‘A’0‘F’00‘X’011111010101101101110100101, закодоване за адаптивним алгоритмом Хаффмана. Обчислити довжини кодів стиснутого і нестиснутого повідомлення в бітах.

4.10 Словникові алгоритми стиснення даних

4.10.1 Основні поняття

Словникові методи стиснення даних передбачають заміну повторюваних підрядків вихідного повідомлення на місце їх появи у цьому повідомленні. Декодування стиснутого повідомлення здійснюється заміною покажчика готовою фразою із словника, на яку цей покажчик вказує.

Найвідомішими словниковими методами є LZ-методи (за прізвищами авторів Ziv та Lempel). Перевагами LZ-методів є високий степінь стиснення даних і швидкість кодування.

Словникові методи поділяють на алгоритми з використанням . «ковзного» вікна і алгоритми із застосуванням словника фраз.

Для алгоритмів з використанням «ковзного» за повідомленням вікна виділяють дві нерівні за об'ємом частини: першу, більшу за розміром, яка включає фрагмент повідомлення, що вже проглянуто; другу, набагато меншу частину вікна, що виступає у якості буфера, що містить ще незакодовані

символи вхідного потоку. Першу частину використовують як словник. Зазвичай об'єм ковзного вікна складає кілька кілобайтів, а розмір буфера - не більше 100 байтів. Алгоритми цієї групи передбачають пошук у словнику (більшій частині вікна) ланцюжків символів, що збігаються із вмістом буфера, і заміну цих ланцюжків покажчиками на їхнє попереднє входження у повідомлення, тобто на вміст словника. Словник в неявному вигляді міститься у закодованих даних, а зберігаються покажчики на повторювані ланцюжки символів (підрядки), що зустрічаються у повідомленні.

Характерними представниками алгоритмів першої групи є LZ77 та модифікований алгоритм LZSS.

Алгоритми із застосуванням словника фраз доповнюють початковий словник джерела словником фраз, що є повторюваними у повідомленні комбінаціями символів початкового словника. При цьому розмір слова збільшується, і для його кодування потрібне більше число біт, але значна частина слова подає не окремі букви, а сполучення букв або цілі слова. Якщо кодер знаходить фразу, що раніше зустрічалаася, він замінює її індексом цієї фрази у словнику. Довжина коду індексу є набагато меншою довжини коду незакодованого підрядка.

Базовою реалізацією таких алгоритмів є алгоритм LZ78, а удосконаленим є алгоритм LZW.

4.10.2 Алгоритм LZ77

За алгоритмом LZ77 друге і подальші входження деякого підрядка символів у повідомленні замінюються покажчиками на його перше або попереднє входження. Алгоритм використовує частину повідомлення, що вже проглянуто, як словник. Для стиснення алгоритм робить спробу замінити наступну фразу повідомлення покажчиком на вміст словника.

З N символів розміра «ковзного» вікна та F - розміру буфера, перші N-F символів є вже закодовані символи, що містить словник, а останні F символів – вміст випереджуючого буфера.

При кодуванні вмісту буфера серед попередніх N-F символів, тобто у словнику, шукається найдовший підрядок, що збігається з початком буфера. Знайдений найбільший збіг кодується тріадою $\langle i, j, a \rangle$, де i - зсув у словнику підрядка, що збігається із початком буфера; j - довжина підрядка, що збігається; a - перший символ, що йде за підрядком, що збігається. Далі алгоритм виконує зсув усього вмісту вікна на $j+1$ символів і водночас читає стільки ж символів вхідного потоку у буфер.

Об'єм пам'яті, що потребує алгоритм-кодер або декодер, визначається розміром вікна N. Довжина коду обчислюється так: довжина підрядка, що співпадає із вмістом словника, не може бути більше розміру буфера F, а зсув цього підрядка у словнику не може бути більше розміру словника мінус 1. Отже, довжина двійкового коду зсуву i буде округлений до більшого цілого $\lceil \log_2(N - F) \rceil$, а довжина коду довжини підрядка j буде округлений у більшу сторону $\lceil \log_2(F + 1) \rceil$, а символ a кодується 8 бітами за таблицею ASCII+.

При декодуванні порядок роботи з вікном подібний до кодуванню. Але на відміну від пошуку підрядків, що збігаються, вони копіюються декодером з вікна згідно з черговою тріадою коду $\langle i, j, a \rangle$.

Приклад 4.11 Закодувати за алгоритмом LZ77 рядок «КРАСНАЯ КРАСКА» з розміром словника 8 байтів, буфера 5 байтів.

Таблицю кодування подано у табл. 4.12.

Довжина отриманого коду Lcode=9·(3+3+8)=126 біт проти LASCI+=14·8=112 біт коду нестисненого рядка.

Декодування (роздавання) за алгоритмом LZ77 за довжини словника 8 байтів стисненого повідомлення:

`<0,0,'K'> <0,0,'P'> <0,0,'A'> <0,0,'C'> <0,0,'H'> <5,1,'Я'> <0,0,' '>`
`<0,4,'K'> <0,0,'A'>` подано у табл. 4.13.

Таблиця 4.12 – Кодування рядка

| Словник (8 Бт) | | | | | | | | Буфер (5 Бт) | | | | | Код | |
|----------------|---|---|---|---|---|---|---|--------------|---|---|---|---|--------------------------------|--------------------------------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 1 | 2 | 3 | 4 | 5 | | |
| . | . | . | . | . | . | . | . | K | P | A | C | H | <code><0, 0, 'K'></code> | |
| . | . | . | . | . | . | . | . | K | P | A | C | H | A | <code><0, 0, 'P'></code> |
| . | . | . | . | . | . | . | . | K | P | A | C | H | A | <code><0, 0, 'A'></code> |
| . | . | . | . | . | . | . | . | K | P | A | C | H | A | <code><0, 0, 'C'></code> |
| . | . | . | . | . | . | . | . | K | P | A | C | H | A | <code><0, 0, 'H'></code> |
| . | . | . | . | K | P | A | C | H | A | Я | | | K | <code><5, 1, 'Я'></code> |
| . | . | . | . | K | P | A | C | H | A | Я | | | K | <code><0, 0, ' '></code> |
| . | K | P | A | C | H | A | Я | | K | P | A | C | | <code><0, 4, 'K'></code> |
| K | P | A | C | H | A | Я | | K | P | A | C | K | | <code><0, 0, 'A'></code> |
| A | Я | K | P | A | C | K | | A | . | . | . | . | . | |

Таблиця 4.13 - Декодування повідомлення

| Вхідний код | Вихід | Словник | | | | | | | |
|--------------------------------|----------------------|---------|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| <code><0, 0, 'K'></code> | <code>«K»</code> | . | . | . | . | . | . | . | K |
| <code><0, 0, 'P'></code> | <code>«P»</code> | . | . | . | . | . | . | K | P |
| <code><0, 0, 'A'></code> | <code>«A»</code> | . | . | . | . | . | K | P | A |
| <code><0, 0, 'C'></code> | <code>«C»</code> | . | . | . | . | K | P | A | C |
| <code><0, 0, 'H'></code> | <code>«H»</code> | . | . | . | K | P | A | C | H |
| <code><5, 1, 'Я'></code> | <code>«Я»</code> | . | K | P | A | C | H | A | Я |
| <code><0, 0, ' '></code> | <code>« »</code> | K | P | A | C | H | A | Я | |
| <code><0, 4, 'K'></code> | <code>«KPACK»</code> | A | Я | K | P | A | C | K | |
| <code><0, 0, 'A'></code> | <code>«A»</code> | Я | K | P | A | C | K | A | |

4.10.3 Алгоритм LZSS

Алгоритм LZSS є модифікацією алгоритму LZ77. Робота алгоритму починається однобітовим префіксом, що відділяє код підрядка, що збігається, від незакодованого символу. Код складається з пари $\langle i, j \rangle$ - зсуву і у словнику підрядка, що збігається з початком буфера, і довжини j цього підрядка. Вікно зсувається на довжину знайденого підрядка або на 1, якщо входження підрядка буфера у словнику не знайдено.

Довжина підрядка у алгоритмі LZSS завжди більше 0 і не може перевищувати розмір буфера F . Тому довжина j двійкового коду довжини

підрядка, що збігається, є округлений до більшого цілого $\lceil \log_2(F) \rceil$, а довжина коду зсуву i – округлений до більшого цілого $\lceil \log_2(N - F) \rceil$.

Приклад 4.12 Закодувати за алгоритмом LZSS рядок «КРАСНАЯ КРАСКА» з розміром словника 8 байтів і буфера 5 байтів.

Результат кодування подано у табл. 4.14.

Довжина отриманого коду Lcode = $\sum_{i=1}^{11} l_i = 7 \cdot 9 + 4 \cdot 7 = 91$ біт.

Довжина нестиснутого повідомлення LASCII+=14·8=112 біт.

Таблиця 4.14 - Кодування повідомлення

| Словник (8 Бт) | | | | | | | | Буфер (5 Бт) | | | | | Код | l_i |
|----------------|---|---|---|---|---|---|---|--------------|---|---|---|---|--------|-------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 1 | 2 | 3 | 4 | 5 | | |
| . | . | . | . | . | . | . | . | К | Р | А | С | Н | 0'К' | 9 |
| . | . | . | . | . | . | . | . | Р | А | С | Н | А | 0'Р' | 9 |
| . | . | . | . | . | . | . | К | Р | А | С | Н | Я | 0'А' | 9 |
| . | . | . | . | . | К | Р | А | С | Н | А | Я | | 0'С' | 9 |
| . | . | . | . | К | Р | А | С | Н | А | Я | | К | 0'Н' | 9 |
| . | . | . | К | Р | А | С | Н | А | Я | | К | Р | 1<5,1> | 7 |
| . | К | Р | А | С | Н | А | Я | Я | | К | Р | А | 0'Я' | 9 |
| К | Р | А | С | Н | А | Я | | | К | Р | А | С | 0' ' | 9 |
| Н | А | Я | | К | Р | А | С | К | А | . | . | . | 1<0,4> | 7 |
| А | Я | | К | Р | А | С | К | А | . | . | . | . | 1<4,1> | 7 |
| | | | | | | | | | | | | | 1<0,1> | 7 |

Приклад 4.14 Декодувати (розпаковувати) повідомлення

0'К'0'Р'0'А'0'С' 0'Н'1<5, 1>0'Я'0' ' 1<0, 4>1<4, 1>1<0, 1>

Результат декодування подано у табл. 4.15.

Таблиця 4.15 – Декодування повідомлення

| Вхідний код | Вихід | Словник | | | | | | | |
|-------------|-------|---------|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0'К' | «К» | . | . | . | . | . | . | К | |
| 0'Р' | «Р» | . | . | . | . | . | . | К | Р |
| 0'А' | «А» | . | . | . | . | . | К | Р | А |
| 0'С' | «С» | . | . | . | . | К | Р | А | С |
| 0'Н' | «Н» | . | . | . | К | Р | А | С | Н |
| 1<5, 1> | «А» | . | . | К | Р | А | С | Н | А |
| 0'Я' | «Я» | . | К | Р | А | С | Н | А | Я |
| 0' ' | « » | К | Р | А | С | Н | А | Я | |
| 1<0, 4> | «КРА» | Н | А | Я | | К | Р | А | С |
| 1<4, 1> | «К» | А | Я | | К | Р | А | С | К |
| 1<0, 1> | « » | Я | | К | Р | А | С | К | А |

4.10.4 Алгоритм LZ78

Алгоритм не використовує вікно, а зберігає словник із фраз повідомлення, що вже проглянуто. Спочатку словник містить лише один порожній рядок. Алгоритм читає символи повідомлення доти, доки накопичуваний підрядок повністю збігається з однією із фраз словника. Як тільки зчитаний підрядок не відповідає фразі словника, алгоритм генерує код з індексу фрази у словнику, що до останнього зчитаного символу містила вхідний підрядок, і символу, що порушив збіг. Потім зчитаний новий підрядок заноситься у словник, і пошук збігу наступного підрядка з однією із фраз словника починається знову. Якщо словник вже заповнений, то з нього видаляють фразу, що найменше використовується у порівняннях.

Для визначення розміру кодів ключовим є розмір словника, оскільки кожний код за алгоритмом LZ78 містить номер фрази у словнику. Звідси випливає, що коди алгоритму мають постійну довжину, яка дорівнює округленому до більшого цілого $\lceil \log_2([\text{розмір словника}]) \rceil + 8$, де 8 - кількість біт для кодування наступного символу, що йде за підрядком, що збігається, за таблицею ACSII+.

Приклад 4.15 Закодувати за алгоритмом LZ78 рядок «КРАСНАЯ КРАСКА» зі словником з 16 фраз.

Результат кодування подано у табл. 4.16.

Покажчик на будь-яку фразу словника розміром 16 фраз – це ціле число від 0 до 15, для кодування якого двійковим рівномірним кодом потрібно 4 біт. Отже, довжина отриманого коду повідомлення $L\text{code}=10 \cdot (4+8)=120$ біт.

Приклад 4.16 Декодувати повідомлення $<0, 'K'><0, 'P'> <0, 'A'><0, 'C'> <0, 'H'><3, 'Я'><0, ' ' ><1, 'P'><3, 'C'><1, 'A'$ за алгоритмом LZ78 для словника з 16 фраз.

Результати декодування подано у табл. 4.17.

Таблиця 4.16 – Кодування повідомлення

| <i>Вхідна фраза (словник)</i> | <i>Код</i> | <i>Індекс фрази</i> |
|-------------------------------|------------|---------------------|
| «» | | 0 |
| «К» | <0, 'K'> | 1 |
| «Р» | <0, 'P'> | 2 |
| «А» | <0, 'A'> | 3 |
| «С» | <0, 'C'> | 4 |
| «Н» | <0, 'H'> | 5 |
| «АЯ» | <3, 'Я'> | 6 |
| « » | <0, ' '> | 7 |
| «KP» | <1, 'P'> | 8 |
| «AC» | <3, 'C'> | 9 |
| «KA» | <1, 'A'> | 10 |

Таблиця 4.17 – Декодування повідомлення

| <i>Вхідний код</i> | <i>Вихід (словник)</i> | <i>Індекс фрази</i> |
|--------------------|------------------------|---------------------|
| | «» | 0 |
| <0, 'K'> | «K» | 1 |
| <0, 'P'> | «P» | 2 |
| <0, 'A'> | «A» | 3 |
| <0, 'C'> | «C» | 4 |
| <0, 'H'> | «H» | 5 |
| <3, 'Я'> | «АЯ» | 6 |
| <0, ' '> | « » | 7 |
| <1, 'P'> | «KP» | 8 |
| <3, 'C'> | «AC» | 9 |
| <1, 'A'> | «KA» | 10 |

4.10.5 Алгоритм LZW

Алгоритм LZW є модифікацією LZ78. Алгоритм починає роботу зі словника розміром 4К, що містить за адресами від 0 до 255 посилання на окремі символи (таблиця ASCII+), а від 256 до 4095 - посилання на підрядки завдовжки більше одного символу. У процесі кодування текст розбивається на підрядки, де кожний новий підрядок є найдовшою фразою, що збігається серед вже проглянутих, плюс один символ. Новий підрядок кодують індексом його префікса (фрази словника) і символом, що порушив збіг, з подальшим занесенням нової фрази у словник (таблицю рядків). При переповненні

словника з нього видаляється або найменш уживана фраза, або усі підрядки завдовжки більше одного символу.

LZW-коди мають постійну довжину, що визначається округленим до більшого цілого $\lceil \log_2([\text{розмір словника}]) \rceil$.

Приклад 4.17 Закодувати за алгоритмом LZW рядок «КРАСНАЯ КРАСКА» зі словником з 500 фраз

Результат кодування подано у табл. 4.18.

Таблиця 4.18 – Кодування повідомлення

| <i>Вхідна фраза wK (словник)</i> | <i>Код w</i> | <i>Індекс фрази</i> |
|--------------------------------------|--------------|---------------------|
| <i>ASCII+</i> | | <i>0 – 255</i> |
| «КР» | 0'К' | 256 |
| «РА» | 0'Р' | 257 |
| «АС» | 0'А' | 258 |
| «ЧН» | 0'С' | 259 |
| «НА» | 0'Н' | 260 |
| «АЯ» | 0'А' | 261 |
| «Я » | 0'Я' | 262 |
| « К» | 0' , | 263 |
| «КРА» | <256> | 264 |
| «ACK» | <258> | 265 |
| «КА» | 0'К' | 266 |
| «А» | 0'А' | |

Довжина отриманого коду $L_{\text{code}} = 12 \cdot \lceil \log_2(500) \rceil = 108$ біт.

LZW-декодер, як і кодер, заносить нові фрази у словник кожного разу, коли знаходить у вхідному потоці новий код. Наприкінці декодування декодер має такий самий словник фраз, що був накопичений кодером при кодуванні. При декодуванні за алгоритмом LZW потрібно дотримуватися правила: словник поповнюється тільки після зчитування першого символу фрази, що розкодовується із наступного за поточним коду.

Приклад 4.18 Декодувати повідомлення 0'К'0'Р'0'А'0'С' 0'Н'0'А'0'Я'0' '<256><258>0'К'0'А', для словника з 500 фраз подано у табл. 4.19.

Особливістю словниковых способів стиснення є те, що словник нових фраз передається декодеру власне без самої передачі, оскільки міститься у неявному вигляді у отриманих кодах.

Таблиця 4.19 – Декодування повідомлення

| <i>Вхідний код</i> | <i>Вихід</i> | <i>Словник</i> | <i>Індекс фрази</i> |
|--------------------|--------------|-----------------------|-----------------------|
| 0'K' | «K» | <i>ASCII+</i> «KP» | <i>0 – 255</i> 256 |
| 0'R' | «P» | «PA» | 257 |
| 0'A' | «A» | «AC» | 258 |
| 0'C' | «C» | «CH» | 259 |
| 0'H' | «H» | «HA» | 260 |
| 0'A' | «A» | «AY» | 261 |
| 0'Я' | «Я» | «Я » | 262 |
| 0' ' | « » | « К» | 263 |
| <256> | «KP» | «КРА» | 264 |
| <258> | «AC» | «ACK» | 265 |
| 0'K' | «K» | «КА» | 266 |
| 0'A' | «A» | «А» | |

4.11 Задачі та завдання за словниковими алгоритмами

- 1 Закодувати повідомлення ACCABCDAACCCCDBB, використовуючи словникові алгоритми стиснення LZ77 і LZSS (словник - 10 байтів, буфер – 5 байти). Обчислити довжину у бітах отриманих кодів.
- 2 Закодувати повідомлення INTERESSIEREN, використовуючи словникові алгоритми стиснення LZ77 і LZSS (словник - 11 байтів, буфер – 3 байти). Обчислити довжину у бітах отриманих кодів.
- 3 Закодувати повідомлення СЛАВА УКРАЇНІ ГЕРОЯМ СЛАВА, використовуючи словникові алгоритми стиснення LZ77 і LZSS (словник - 14 байтів, буфер – 5 байти). Обчислити довжину у бітах отриманих кодів.

5 АЛГОРИТМИ ПОБУДОВИ ЗАВАДОСТІЙКИХ КОДІВ

5.1 Лінійні блокові коди

Кодер для блокових кодів перетворює інформаційні блоки завдовжки k символів у повідомлення з n символів, з яких $r=n-k$ є надлишковими (перевірними, контрольними). Перевірні символи забезпечують виявлення й виправлення помилок, спричинених завадами у каналі зв'язку. Кожний блок з n символів залежить лише від відповідного k -символьного інформаційного блоку і не залежить від інших блоків.

Кодер для згорткових кодів перетворює послідовність з k інформаційних символів у блок з n кодових символів, де $n>k$. Кодовий n -символьний блок залежить не тільки від k -символьного інформаційного блоку, наявного на вході у поточний момент часу, але і від попередніх m блоків повідомлення. Згорткові коди часто використовують для побітового передавання даних каналами.

Блоковий код завдовжки n символів, що складається з 2^k кодових слів, називають лінійним (k, n) -кодом за умови, що всі 2^k кодових слів утворюють k -вимірний підпростір векторного простору n -послідовностей двійкового поля $GF(2)$, тобто порозрядна сума за модулем $2 \pmod{2}$ двох кодових слів також є кодовим словом даного коду. Систематичний код має інформаційну частину з k символів і надлишкову (перевірну) частину з $n-k$ символів постійної довжини. Найпростішим лінійним блоковим кодом є $(n-1, n)$ - код з контролем парності з використанням перетворення виду $E(m_1, \dots, m_k) = (m_1, \dots, m_k, m_{k+1})$, де m_{k+1} є символом парності або непарності інформаційних розрядів m_1, \dots, m_k .

Помилки у бітах для двійкових симетричних каналів вважають рівномовірними й незалежними і оцінюють як: p - ймовірність безпомилкової передачі біта і $q = 1-p$ - ймовірність помилкової передачі повідомлення. Тому

ймовірність передачі n бітів з k помилками визначається за формулою Бернуллі

$$P_n(k) = C_n^k q^k p^{n-k}, \text{ де } C_n^k = \frac{n!}{k!(n-k)!} - \text{кількість сполук з } n \text{ елементів по } k$$

5.2 Ітеративний код

Ітеративний код є найпростішим кодом з виправленням помилок. Повідомлення та розміщують у вигляді матриці з додаванням до кожного рядка і стовпця контрольних символів перевірки на парність. За умови виникнення однієї помилки перевірка на парність у відповідному рядку і стовпці не виконуватиметься, а координати помилки однозначно визначаються номерами стовпця і рядка, в яких не виконується перевірка на парність.

Приклад 5.1. Для кодування послідовності з $k=12$ інформаційних елементів застосовується ітеративний метод. Записати твірну матрицю еквівалентного лінійного блокового коду. Закодувати повідомлення (110111000110) . Виправити помилку в прийнятій послідовності (11011110001110001111) . Визначити надлишковість коду.

Для розв'язання задачі інформаційну послідовність розмістимо у вигляді матриці розмірністю 3×4 . Вихідна матриця кодування для інформаційної послідовності $(m_1, m_2, \dots, m_{12})$ має вигляд

$$\begin{pmatrix} m_1 & m_2 & m_3 & m_4 \\ m_5 & m_6 & m_7 & m_8 \\ m_9 & m_{10} & m_{11} & m_{12} \end{pmatrix} \Rightarrow \begin{pmatrix} m_1 & m_2 & m_3 & m_4 & r_1 \\ m_5 & m_6 & m_7 & m_8 & r_2 \\ m_9 & m_{10} & m_{11} & m_{12} & r_3 \\ r_4 & r_5 & r_6 & r_7 & r_8 \end{pmatrix},$$

де r_1, r_2, \dots, r_8 – перевірні елементи.

Тоді послідовність, що зберігається або передається по каналу зв'язку, є: $u = (m_1, m_2, m_3, m_4, r_1, m_5, m_6, m_7, m_8, r_2, m_9, m_{10}, m_{11}, m_{12}, r_3, r_4, r_5, r_6, r_8)$.

З вихідної матриці отримуємо систему перевірних рівнянь, що визначає правила знаходження перевірних елементів:

$$\begin{cases} r_1 = m_1 + m_2 + m_3 + m_4, \\ r_2 = m_5 + m_6 + m_7 + m_8, \\ r_3 = m_9 + m_{10} + m_{11} + m_{12}, \\ r_4 = m_1 + m_5 + m_9, \\ r_5 = m_2 + m_6 + m_{10}, \\ r_6 = m_3 + m_7 + m_{11}, \\ r_7 = m_4 + m_8 + m_{12}, \\ r_8 = m_1 + m_2 + \dots + m_{12}. \end{cases}$$

Для системи перевірних рівнянь знаходимо твірну матрицю $G_{12 \times 20}$ еквівалентного лінійного блокового коду, яку можна легко привести до канонічного вигляду, розмістивши стовпці, що відповідають перевірним елементам у правій частині твірної матриці, а одиничну підматрицю, що визначає інформаційну частину кодового слова – у лівій:

$$G_{12 \times 20} = \left(\begin{array}{cccccccccccc|ccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \begin{matrix} m_1 & m_2 & m_3 & m_4 & r_1 & m_5 & m_6 & m_7 & m_8 & r_2 & m_9 & m_{10} & m_{11} & m_{12} & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 \end{matrix}$$

Закодуємо задану інформаційну послідовність $m=(110111000110)$.

Розмістимо її у вигляді матриці 3×4 і допишемо перевірні елементи до кожного рядка і стовпця:

$$\left| \begin{array}{cccc} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{array} \right| \Rightarrow \left| \begin{array}{cccc|c} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 1 \end{array} \right|.$$

Тобто кодовим слово буде (11011110000110001111) .

За умовою задачі на приймайльній стороні прийнято послідовність (11011110001110001111) . Декодуємо її за ітеративним методом.

Запишемо прийняте слово у вигляді матриці (у даному випадку розмірністю 4×5) і виконаємо перевірку на парність за її кожним рядком і стовпчиком. У разі відсутності помилки контрольні елементи парності за рядками і стовпчиками матриці мають значення 0. Невиконання контролю парності у стовпчику і рядку однозначно визначить координати помилкового елемента – це дозволить його відправити.

$$\left| \begin{array}{ccccc} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{array} \right| \Rightarrow \left| \begin{array}{ccccc|c} 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right|$$

Помилковим є елемент $m[3,1]$, який відправляємо.

$$\left| \begin{array}{ccccc} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ \textcolor{red}{1} & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{array} \right| \Rightarrow \left| \begin{array}{ccccc} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{array} \right|$$

Отже, відправлена комбінація така:

$$y = (11011110000110001111).$$

Декодуємо її, прибравши перевірні елементи:

$$(11011110000110001111) \Rightarrow (110111000110).$$

Надлишковість коду $\rho_k = 1 - \frac{12}{20} = 0,4$.

5.3 Способи подання лінійних кодів

Основними способами подання лінійних кодів є табличний, за допомогою систем перевірних рівнянь, за допомогою твірної матриці,

перевірної матриці. За табличним способом кожній інформаційній послідовності ставиться у відповідність кодове слово з таблиці кодів.

Система перевірних рівнянь визначає правила знаходження перевірних символів залежно від інформаційних, наприклад

$$\begin{cases} r_1 = m_1, \\ r_2 = m_2, \\ r_3 = m_3, \\ r_4 = m_1 + m_2 + m_3; \end{cases}$$

де r_1, r_2, r_3, r_4 - перевірні (контрольні) символи.

Твірна матриця G розміром $k \times n$

$$G_{k \times n} = \left(\begin{array}{cccc|ccccc} 1 & 0 & 0 & \dots & 0 & p_{11} & p_{12} & \dots & p_{1,n-k} \\ 0 & 1 & 0 & \dots & 0 & p_{21} & p_{22} & \dots & p_{2,n-k} \\ 0 & 0 & 1 & \dots & 0 & \dots & \dots & \dots & \dots \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & p_{k1} & p_{k2} & \dots & p_{k,n-k} \end{array} \right).$$

одинична підматриця $I_{k \times k}$ перевірна підматриця $P_{k \times (n-k)}$

лінійного блокового систематичного (k, n) -коду визначає кожне кодове слово як лінійну комбінацію рядків матриці G , а кожну лінійну комбінація рядків G - кодовим словом.

Для блока повідомлення $m = (m_1, m_2, \dots, m_k)$ кодове слово є послідовність $u = m \times G$, де для $i=1, 2, \dots, k$ $u_i = m_i$; для $i=k+1, \dots, n$ $u_i = m_1 p_{1i} + m_2 p_{2i} + \dots + m_k p_{ki}$; $i=1, 2, \dots, n-k$ - номер стовпчика перевірної частини $P_{k \times (n-k)}$ твірної матриці $G_{k \times n}$.

Наприклад, для твірної матриці

$$G_{4 \times 7} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

кодові слова визначають як інформаційні і перевірні розряди

$$u = m \times G = (m_1, m_2, m_3, m_4) \times \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} = (m_1, m_2, m_3, m_4, m_1+m_3+m_4, m_1+m_2+m_3, m_2+m_3+m_4).$$

5.4 Перевірна матриця і кодовий синдром

Перевірна матриця $H_{(n-k) \times n}$ розмірності $(n-k) \times n$ має властивість: для кодового слова u і $u \times H^T = 0$, тобто ортогональна будь-якій кодовій послідовності даного коду і має структуру

$$\underline{H}_{(n-k) \times n} = \left\langle \begin{array}{cccc|cccc} p_{11} & p_{21} & \dots & p_{k1} & 1 & 0 & 0 & \dots & 0 \\ p_{12} & p_{22} & \dots & p_{k2} & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots \\ \dots & \dots \\ p_{1n-k} & p_{2n-k} & \dots & p_{kn-k} & 0 & 0 & 0 & \dots & 1 \end{array} \right\rangle, \quad (1)$$

$\underbrace{P_{(n-k) \times k}}^T \quad \underbrace{I_{(n-k) \times (n-k)}}$

де $P_{(n-k) \times k}^T$ - транспонована перевірна підматриця твірної матриці $G_{k \times n}$;

$I_{(n-k) \times (n-k)}$ - одинична підматриця.

За допомогою перевірної матриці можна визначити, чи є прийнята послідовність кодовим словом даного коду. Наприклад, для матриці кода (4, 7)

$$\underline{H}_{3 \times 7} = \left(\begin{array}{ccc|cc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array} \right).$$

і послідовності $y=(1011001)$ отримаємо

$$y \times \underline{H}^T = (1011001) \times \left(\begin{array}{ccc|cc} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array} \right)^T = (\color{red}1 \ 0 \ 1) \neq 0.$$

Тобто $y=(1011001)$ не є кодовим словом даного коду.

Перевірні матриці використовують для перевірки наявності помилок $e=(e_1, e_2, \dots, e_n)$ у повідомленнях $u=(u_1, u_2, \dots, u_n)$, що надходять до каналу зв'язку. На виході каналу зв'язку з шумами отримують послідовність $y = u+e$, де u - передане кодове слово; e – вектор помилок у каналі. Вектор помилок

$e=(e_1, e_2, \dots, e_n)$, є двійкою послідовністю завдовжки n з одиницями у тих позиціях, де виникли помилки.

Наприклад, вектор помилок $e=(0001000)$ означає однократну помилку у четвертому біті, $e=(1100000)$ - двократну помилку у першому і другому бітах. Для $u=(0001000)$, $e=(0001000)$, $y=(0000000)$.

Для перевірки наявності помилок у прийнятій послідовності y , декодер обчислює кодовий синдром S як $(n-k)$ - послідовність

$$S=(S_1, S_2, \dots, S_{n-k})=y \times H^T,$$

де y - прийнята кодована послідовність; H^T - транспонована перевірна матриця коду.

Послідовність y є кодовим словом при $S=(0\ 0\dots\ 0)$, і не є кодовим словом за умови $S \neq 0$.

Наприклад, для $y=(y_1, y_2, \dots, y_7)$ синдром

$$S=y \times H_{3 \times 7}^T = (y_1, y_2, \dots, y_7) \times \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}^T = ((y_1+y_3+y_4+y_5), (y_1+y_2+y_3+y_6), (y_2+y_3+y_4+y_7)).$$

Декодер не виявляє помилки, для яких синдром $S=y \times H^T=0$.

Кодовий синдром залежить лише від вектора помилок і не залежить від переданого слова, оскільки

$$S=y \times H^T = (u+e) \times H^T = u \times H^T + e \times H^T = 0 + e \times H^T = e \times H^T,$$

За наявним вектором e можна відновити кодове слово: $u^*=y+e$.

Наприклад для $e_4=(0001000)$

$$e_4 \times H^T = (0001000) \times \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}^T = (\textcolor{red}{101}) - \text{помилка у 4-му біті}$$

Помилку можна вилучити за $u^*=y+e$.

Існує однозначна відповідність між координатами поодиноких помилок і їх синдромами, тобто, знаючи синдром, можна визначити позицію коду, в якій виникла помилка.

Наприклад, з перевірної матриці лінійного блокового (4, 7)- коду знайдемо систему перевірних рівнянь:

$$\begin{aligned} \mathbf{u} \times \mathbf{H}_{3 \times 7}^T &= (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_7) \times \mathbf{H}_{3 \times 7}^T = (\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \times \\ &\times \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}^T = (\mathbf{r}_1 + \mathbf{m}_1 + \mathbf{m}_3 + \mathbf{m}_4, \mathbf{r}_2 + \mathbf{m}_1 + \mathbf{m}_2 + \mathbf{m}_3, \\ &\mathbf{r}_3 + \mathbf{m}_2 + \mathbf{m}_3 + \mathbf{m}_4) = (0 \ 0 \ 0). \end{aligned}$$

Звідси

$$\begin{cases} \mathbf{u}_5 = \mathbf{r}_1 = \mathbf{m}_1 + \mathbf{m}_3 + \mathbf{m}_4; \\ \mathbf{u}_6 = \mathbf{r}_2 = \mathbf{m}_1 + \mathbf{m}_2 + \mathbf{m}_3; \\ \mathbf{u}_7 = \mathbf{r}_3 = \mathbf{m}_2 + \mathbf{m}_3 + \mathbf{m}_4; \end{cases}$$

де $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4)$ – інформаційні символи; $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ - перевірні символи; $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_7) = (\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ - кодове слово.

Виникнення помилки у кодовій комбінації призведе до невиконання тих рівнянь у системі перевірних рівнянь коду, в які входить значення помилкового розряду.

Наприклад, якщо помилка у четвертому розряді кодової комбінації $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4, \mathbf{u}_5, \mathbf{u}_6, \mathbf{u}_7) = (\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$, то не виконаються перше і третє рівняння, в які входить помилковий символ $\mathbf{u}_4 = \mathbf{m}_4$. Вектором синдрому буде послідовність $S = (101)$, що збігається з четвертим стовпцем перевірної матриці коду \mathbf{H} .

У такий спосіб номер помилкового розряду кодової комбінації є номером стовпчика перевірної матриці \mathbf{H} , що збігається з вектором синдрому. Це дозволяє визначити місце виникнення помилки і таким чином її віправити.

Кодовий синдром виправляє тільки однократні помилки в коді.
Наприклад, $e=(0100010)$ і

$$S = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}^T \times \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} = (001).$$

Цей синдром збігається з сьомим стовпцем перевірної матриці H , що означає наявність однієї помилки у сьому біті. Отже, декодер не тільки не виправить помилкові біти, але і внесе помилку у позицію, де її не було. Тобто лінійний блоковий $(4, 7)$ - код не виправляє двохкратні помилки і помилки більшої кратності.

Основними правилами побудови твірної матриці лінійного блокового (k, n) -коду є:

- 1) кількість початкових кодових комбінацій (число рядків) твірної матриці дорівнює k , тобто кількості інформаційних елементів;
- 2) усі кодові комбінації твірної матриці повинні відрізнятися, але нульова комбінація до їх складу не входить;
- 3) усі кодові комбінації твірної матриці лінійно незалежні;
- 4) кількість одиниць в кожній кодовій комбінації твірної матриці повинна бути не меншою за d_{min} ;
- 5) кодова відстань між будь-якою парою кодових слів повинна бути не меншою за d_{min} .

5.5 Поліноміальне кодування інформації. Циклічні коди

Лінійний блоковий (k, n) -код можна подати коефіцієнтами полінома

$$u(x) = u_0 + u_1 x + u_2 x^2 + \dots + u_{n-1} x^{n-1},$$

В полі двійкових символів $GF(2)$ суму двох поліномів $f(x)$ і $g(x)$ визначають як поліном з $GF(2)$

$$f(x) + g(x) = \sum_{i=0}^{n-1} (f_i + g_i)x^i,$$

тобто додавання виконують як операцію додавання за модулем два коефіцієнтів при одинакових степенях x . Добуток двох поліномів $f(x)$ і $g(x)$ в полі двійкових символів $GF(2)$ є поліномом з $GF(2)$

$$f(x) \cdot g(x) = \sum_{i=0}^{n-1} \left(\sum_{j=0}^i f_j g_{i-j} \right) x^i,$$

тобто знаходять як добуток поліномів за правилом множення степеневих функцій, а коефіцієнти однієї степені додаються за модулем два.

Наприклад:

$$\begin{array}{r} x^3 + x^2 + 0 \cdot x + 1 \\ \times \quad x^2 + x \\ \hline x^4 + x^3 + 0 \cdot x^2 + x \\ + \quad x^5 + x^4 + 0 \cdot x^3 + x^2 \\ \hline x^5 + 0 \cdot x^4 + x^3 + x^2 + x = x^5 + x^3 + x^2 + x. \end{array}$$

Ділення поліномів виконують за правилами ділення степеневих функцій, а операція віднімання замінюють додаванням за модулем два.

Для кожної пари поліномів $c(x)$ і $d(x)$ ($d(x) \neq 0$) існує єдина пара поліномів $q(x)$, $\rho(x)$, для яких

$$c(x) = q(x) \cdot d(x) + \rho(x),$$

де $q(x)$ – частка, $\rho(x)$ - залишок від ділення

Наприклад:

$$\begin{array}{r} x^4 + 0 \cdot x^3 + x^2 + x + 1 \quad | \quad x + 1 \\ + \quad x^4 + x^3 \\ \hline x^3 + x^2 + x + 1 \\ + \quad x^3 + x^2 \\ \hline x + 1 \\ + \quad x + 1 \\ \hline 0 - \text{залишок } \rho(x). \end{array}$$

Поліноміальним кодом називають множину всіх многочленів степені не більше $n-1$, що мають спільний множник – деякий фіксований многочлен $g(x)$ степеня $r=n-k$. Цей многочлен $g(x)$ називають твірним многочленом коду.

Поліноміальний код з твірним многочленом $g(x)$ кодує повідомлення $m(x)$ поліномом

$$u(x) = m(x) \cdot g(x) = u_0 + u_1x + u_2x^2 + \dots + u_{n-1}x^{n-1},$$

або кодовим словом з коефіцієнтів цього многочлена $u = (u_0, u_1, \dots, u_{n-1})$.

Матриця $G_{k \times n}$ поліноміального коду з твірним многочленом $g(x)$ степеня $r=n-k$ має вигляд

$$G_{k \times n} = \begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_r & \dots & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_{r-1} & g_r & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots & g_r \end{pmatrix},$$

Ненульові елементи в i -му рядку є послідовністю коефіцієнтів твірного многочлена, розташованих з j -го по $(j+r)$ -й стовпець.

Приклад 5.2 Поліноміальний $(5, 8)$ - код задано твірним многочленом $g(x)=1+x^2+x^3$. Закодуємо за його допомогою повідомлення (01011) .

Повідомлення (01011) можна подати многочленом

$$m(x) = 0 \cdot x^0 + 1 \cdot x + 0 \cdot x^2 + 1 \cdot x^3 + 1 \cdot x^4 = x + x^3 + x^4$$

Тоді кодовим словом буде поліном

$u(x) = m(x) \cdot g(x) = (x + x^3 + x^4) \cdot (1 + x^2 + x^3) = x + x^3 + x^4 + x^3 + x^5 + x^6 + x^4 + x^6 + x^7 = x + (1+1)x^3 + (1+1)x^5 + (1+1)x^6 + x^7 = x + x^5 + x^7$. Цьому поліному відповідає кодова послідовність $u = (01000101)$.

Поліноміальний $(5, 8)$ - код можна також подати за допомогою твірної матриці для твірного полінома $g(x)=1+x^2+x^3$

$$G_{5 \times 8} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

або відображення: $00000 \rightarrow 00000000$, $00001 \rightarrow 00001011$, $00010 \rightarrow 00010110$, $01011 \rightarrow 01000101$.

Вектор помилок $e=e_0, \dots, e_{n-1}$ є не визначеним у тому і лише у тому

випадку, коли його многочлен $e(x) = e_0 + e_1x + \dots + e_{n-1}x^{n-1}$ ділиться на твірний поліном коду $g(x)$ без остачі. Тобто прийнята послідовність $c(x) = m(x)g(x) + e(x)$ ділиться на $g(x)$ без остачі тоді і тільки тоді, коли $e(x)$ ділиться на $g(x)$ без остачі.

Тому помилка, многочлен якої не ділиться на $g(x)$ можна визначити, а помилка, многочлен якої ділиться на $g(x)$, не можна визначити. Як наслідок, виявлення помилки поліноміальним кодом з твірним поліномом $g(x)$ може бути здійснене за допомогою ділення многочленів. Якщо залишок від ділення многочлена прийнятої послідовності на твірний поліном $g(x)$ ненульовий, то при передачі відбулося спотворення даних.

Якщо твірний поліном $g(x)$ не є дільником жодного з многочленів вигляду x^{j+1} , де $j < n$, то мінімальна відстань між кодовими словами відповідного поліноміального коду $d_{min} \geq 3$.

До простих і поширеніх поліноміальних кодів відносять циклічні коди. Лінійний блоковий (k, n) - код називають циклічним, якщо в результаті циклічного зсуву кодового слова $u = (u_0, u_1, \dots, u_{n-1})$ отримують нове кодове слово $v = (u_{n-1}, u_0, u_1, \dots, u_{n-2})$, яке також належить даному коду.

Особливості застосування циклічних поліноміальних кодів визначають такі їх властивості: кожний ненульовий поліном повинен мати степінь в межах від $(n-k)$ до $n-1$; існує лише один кодовий поліном $g(x) = 1 + g_1x + g_2x^2 + \dots + g_{n-k-1}x^{n-k-1} + x^{n-k}$ степені $(n-k)$, що є дільником кожного кодового полінома $u(x) = m(x) \cdot g(x)$ і який називають твірним поліномом коду.

Кодове слово циклічного коду складається з перевірної частини з $(n-k)$ перевірних символів і інформаційної частини m завдовжки k символів. Перевірні символи є коефіцієнтами полінома $\rho(x)$ – остачі від ділення кодового слова $u(x) = m(x) \cdot x^{n-k}$ на твірний поліном $g(x)$.

Алгоритм побудови циклічного (k, n) -коду для послідовності $m = (m_0, m_1,$

m_2, \dots, m_{k-1}) такий:

- 1) многочлен інформаційної послідовності $m(x)$ домножають на x^{n-k} , що забезпечує зсув праворуч на $n-k$ розрядів;
- 2) отриманий поліном ділять на твірний поліном коду $g(x)$;
- 3) остача від ділення $x^{n-k} \cdot m(x)$ на $g(x)$ додається до $x^{n-k} \cdot m(x)$, тобто записується у молодших $n-k$ розрядах.

Важлива властивість циклічного (k, n) -коду є те, що твірний поліном $g(x)$ ділить без остачі двочлен x^{n+1} , тобто комбінації лінійного коду мають властивість циклічності за виконання умови

$$x^n + 1 = g(x) \cdot h(x).$$

Двочлен x^{n+1} може бути розкладений на кілька незвідних поліномів, тобто поліномів, які не можуть бути подані як добуток многочленів меншого степеня і діляться або самі на себе, або на 1.

У якості твірних поліномів циклічних кодів використовують незвідні поліноми і їх добутки, оскільки вони є дільниками двочлена $x^n + 1$. Поширеними є такі твірні поліноми (табл. 5.1).

Поліном $h(x)$ степеня k , що є часткою від ділення двочлена x^{n+1} на твірний поліном коду $g(x)$, називають перевірним поліномом. Оскільки $h(x)$ однозначно зв'язаний з $g(x)$, то він також визначає код. Це дозволяє дати інше визначення циклічного лінійного кода і синдрома.

Циклічним називають лінійний (k, n) -код, усі 2^k кодові комбінації якого подані поліномами степеня $n-1$ і менше, які діляться без остачі на деякий поліном $g(x)$ степеня $r=n-k$, що є дільником двочлена x^{n+1} .

Для канала зв'язку позначимо через $u(x)$ та $y(x)$ поліноми, що відповідають переданому кодовому слову u і прийнятій послідовності y .

Розділивши поліном прийнятого повідомлення $y(x)$ на твірний поліном коду $g(x)$, отримаємо поліноми частки від ділення $q(x)$ і остачі $s(x)$, тобто

$$y(x) = q(x)g(x) + s(x).$$

Таблиця 5.1-Твірні поліноми

| <i>Твірний поліном $g(x)$</i> | <i>Двійковий запис полінома</i> |
|---|--|
| $1+x$ | 1 1 |
| $1+x+x^2$ | 1 1 1 |
| $1+x+x^3$ | 1 1 0 1 |
| $1+x^2+x^3$ | 1 0 1 1 |
| $1+x+x^4$ | 1 1 0 0 1 |
| $1+x^3+x^4$ | 1 0 0 1 1 |
| $1+x+x^2+x^4$ | 1 1 1 0 1 |
| $1+x^2+x^3+x^4$ | 1 0 1 1 1 |
| $1+x+x^2+x^3+x^4$ | 1 1 1 1 1 |
| $1+x^2+x^3+x^5$ | 1 0 1 0 0 1 |
| $1+x^3+x^5$ | 1 0 0 1 0 1 |
| $1+x+x^2+x^3+x^5$ | 1 1 1 1 0 1 |
| $1+x+x^2+x^4+x^5$ | 1 1 1 0 1 1 |
| $1+x^4+x^5+x^6$ | 1 0 0 0 1 1 1 |
| $1+x+x^2+x^5+x^6+x^7+x^8$ | 1 1 1 0 0 1 1 1 1 |
| $1+x^3+x^5+x^9$ | 1 0 0 1 0 1 0 0 0 1 |
| $1+x+x^5+x^{10}+x^{11}$ | 1 1 0 1 0 0 1 0 0 0 1 1 |
| $1+x+x^2+x^3+x^5+x^7+x^8+x^{11}$ | 1 1 1 1 0 1 0 1 1 0 0 1 |
| $1+x^3+x^4+x^6+x^8+x^9+x^{10}+x^{11}$ | 1 0 0 1 1 0 1 0 1 1 1 1 |
| $1+x+x^2+x^3+x^4+x^{12}+x^{13}+x^{14}+x^{15}$ | 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 |
| $1+x^5+x^{12}+x^{16}$ | 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 |

Якщо $y(x)$ є кодовим поліномом, то він ділиться на $g(x)$ без остачі, тобто $s(x)=0$. Ненульова остача $s(x)\neq 0$ є ознакою наявності помилки у прийнятій послідовності, тобто $s(x)$ є кодовим синдромом.

Поліноми синдрому $s(x)$ і вектора помилок в каналі $e(x)$ мають вигляд

$$s(x)=s_0+s_1x+\dots+s_{n-k-1}x^{n-k-1}.$$

$$e(x)=e_0+e_1x+\dots+e_{n-1}x^{n-1}$$

Многочлен синдрому однозначно зв'язаний з многочленом помилок $e(x)$, дозволяє не тільки виявляти, але й виправляти помилки у прийнятій послідовності, поліном $y(x)$ якої визначають як $y(x)=u(x)+e(x)$. І $u(x)+y(x)=e(x)$. Оскільки $y(x)=q(x)\cdot g(x) + s(x)$, $u(x)=m(x)\cdot g(x)$, то $e(x)=[m(x)+q(x)]\cdot g(x)+s(x)=f(x)\cdot g(x)+s(x)$, тобто поліном синдрому $s(x)$ – це остача від ділення многочлена помилки $e(x)$ на твірний поліном коду $g(x)$. Тому за поліномом синдрому можна однозначно визначити вектор помилок і виправити помилку.

5.6 Задачі та завдання

1 Перевірна матриця лінійного блокового (3, 6)- коду має вигляд

$$\mathbf{H}_{3 \times 6} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Побудувати твірну матрицю і систему перевірних рівнянь коду. Визначити синдром виправлення однократних помилок в комбінаціях коду. Навести приклад виявлення і виправлення помилки кодом.

2 Перевірна матриця лінійного блокового (3, 6)- коду має вигляд

$$\mathbf{H}_{3 \times 6} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Побудувати твірну матрицю коду. Закодувати повідомлення 101, 001. Використовуючи синдром, визначити помилку (якщо вона є) і декодувати послідовності 011001; 110100; 001111.

3 Лінійний блоковий код заданий твірною матрицею вигляду

$$\mathbf{G}_{3 \times 6} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Побудувати перевірну матрицю коду. Визначити синдром виправлення поодиноких помилок цим кодом. Побудувати таблицю кодів. Навести приклад виявлення і виправлення помилки кодом.

4 Поліноміальний код заданий твірним многочленом $g(x)=1+x+x^5+x^7$. Закодувати повідомлення 0100, 10001101, 11110.

5 Поліноміальний (3, 6) - код заданий многочленом вигляду $g(x)=1+x^2+x^3$. Побудувати твірну матрицю коду і таблицю декодування з виправленням помилки.

6 ОСНОВНА ЛІТЕРАТУРА

1. Шеннон К.-Э. Работы по теории информации и кибернетике / Шеннон К.-Э. – М.: Изд-во иностр. Лит., 1963. 830 с.
2. Коваленко А.Є. Теорія інформації та кодування: Практикум для студентів напряму підготовки 6.040303 «Системний аналіз» / А.Є.Коваленко.- К.:НТУУ «КПІ», 2014.-198 с
3. Тулякова Н.О. Теорія інформації: навчал. Посіб /.. Тулякова Н..О. - Суми: Вид-во СумДУ, 2008.- 212 с..
4. Блейхут Р. Теория и практика кодов, контролирующих ошибки / Блейхут Р. – М.: Мир, 1986.- 576 с.
5. Стратонович Р.Л. Теория информации / Стратонович Р.Л. – М. : Сов. Радио, 1975. – 375 с.
6. Хемминг Р.В. Теория информации и теория кодирования / Хемминг Р.В. – М. Радио и связь, 1983. -412 с.

7 ДОДАТКОВА ЛІТЕРАТУРА

1. Питерсон Р.Коды, исправляющие ошибки / Питерсон Р., Уэлдон Э. – М.: Мир, 1976.- 347 с.
2. Цымбал В.П. Теория информации и кодирование / Цымбал В.П. – К. Вища школа, 1992. – 263 с.
3. Чисар И.. Теория информации / Чисар И., Кerner Я. – М.: Мир, 1985. - 480 с.

Додаток А Приклади питань і варіантів завдань до контрольної роботи 1

Питання до контрольної роботи 1

1. Поняття теорії інформації, кодування. Галузі застосування, напрями.
2. Інформаційна ентропія.. Повідомлення.
3. Властивості ентропії. Вимоги до виміру і функції.
4. Інформаціна ентропія за основою g і 2 . Часткова ентропія.
5. Які існують види інформації?
6. Як перевести неперервну інформацію в дискретний (цифровий) вигляд?
7. Що таке частота дискретизації неперервної інформації?
8. Як формулюється теорема дискретизації?
9. Класифікація кодів.
10. Характеристика системи передавання даних.
11. Способи вимірювання інформації.
12. Інтуїтивні властивості інформації.
13. Кількість інформації взаємозалежних повідомлень
14. Ентропія джерела.
15. Властивості ентропії джерела. Формула Хартлі.
16. Надлишковість джерела.
17. Розподіл сумісних імовірностей десертних випадкових величин. Кількість інформації
18. Властивості кількості інформації і ентропії
19. Задача на знаходження ентропії дискретних випадкових величин.

Приклади варіантів завдань до контрольної роботи 1

Варіант 1

1. Поняття теорії інформації, кодування. Галузі застосування, напрями.
2. Які існують види інформації?
3. Характеристика системи передавання даних.
4. Властивості ентропії джерела. Формула Хартлі.
5. Знайти ентропії ДВВ X, Y, Z та кількість інформації, що містить ДВВ $Z = 2 * X - Y$ стосовно X . X, Y – незалежні ДВВ., які задаються такими розподілами

| | | | | | | |
|---|-----|-----|---|-----|-----|-----|
| X | 1 | 2 | Y | 0 | 1 | 4 |
| p | 1/4 | 3/4 | q | 1/2 | 1/4 | 1/4 |

Варіант 3

1. Інформаційна ентропія.. Повідомлення.
2. Що таке частота дискретизації неперервної інформації?
3. Характеристика системи передавання даних.
4. Способи вимірювання інформації.
5. Знайти ентропії ДВВ X, Y, Z та кількість інформації, що містить ДВВ $Z = X^2 + Y^2$ стосовно X . X, Y – незалежні ДВВ., які задаються такими розподілами

| | | | | | |
|---|-----|-----|---|-----|-----|
| X | 1 | 2 | Y | 1 | 2 |
| p | 1/8 | 7/8 | q | 3/4 | 1/4 |

Додаток Б Приклади питань і варіантів завдань до контрольної роботи 2

Питання до контрольної роботи 2

- 1 Що таке економне кодування інформації? З якою метою воно здійснюється?
- 2 Які існують способи задання кодів?
- 3 Що таке рівномірні й нерівномірні коди?
- 4 З якою метою використовуються оптимальні нерівномірні коди?
- 5 Що таке надлишковість коду? Як вона визначається?
- 6 Яке кодування інформації називається статистичним? Які алгоритми стиснення даних відносять до статистичних?
- 7 Що таке оптимальне кодування інформації? Який критерій оптимальності статистичних кодів?
- 8 Які коди називаються префіксними? Що таке вектор Крафта? Як записується нерівність Крафта? У чому полягає умова оптимальності префіксних кодів?
- 9 У чому полягає алгоритм побудови оптимального коду Шеннона-Фано?
- 10 У чому полягає алгоритм побудови оптимального коду Хаффмана?
- 11 Які переваги та недоліки використання оптимального кодування Шеннона-Фано і Хаффмана?
- 12 Чим визначається верхня границя стиснення інформації? Які існують граници стиснення при використанні оптимального кодування Шеннона-Фано і Хаффмана?
- 13 Які коди характеризуються «наявністю пам'яті»?
- 14 Які коди називаються блоковими? Що таке порядок блокового коду?
- 15 У чому полягає метод блокування повідомлень? Як будеться блоковий код Хаффмана?
- 16 У чому полягає арифметичний алгоритм кодування інформації? Які його переваги в порівнянні з іншими статистичними методами стиснення інформації?
- 17 Як здійснюється декодування даних за арифметичним алгоритмом?
- 18 У чому полягає адаптивний алгоритм Хаффмана? Що таке упорядковане дерево Хаффмана?
- 19 Як здійснюється кодування/декодування вхідних даних за адаптивним алгоритмом Хаффмана?
- 20 Які переваги і недоліки адаптивного алгоритму Хаффмана?
- 21 У чому полягає основна ідея словниковых методів стиснення інформації? У чому переваги використання словниковых методів у порівнянні із статистичними?
- 22 На які основні групи поділяються словникові алгоритми стиснення? Які їх відмітні риси?
- 23 У чому полягають словникові методи стиснення з використанням «ковзного» вікна LZ77, LZSS? Чим визначається довжина кодів цих алгоритмів?
- 24 Які переваги модифікованого алгоритму LZSS у порівнянні з LZ77?
- 25 Які недоліки словниковых алгоритмів LZ77, LZSS?
- 26 У чому полягають словникові методи стиснення LZ78, LZW? Чим визначається довжина кодів для цих алгоритмів? Які переваги модифікації LZW?
- 27 Які переваги алгоритмів LZ78, LZW у порівнянні з LZ77, LZSS?

Приклади варіантів завдань до контрольної роботи 2

Варіант 1

- 1.3 якою метою використовуються оптимальні нерівномірні коди?
- 2.У чому полягає алгоритм побудови оптимального коду Хаффмана?
- 3.У чому полягає метод блокування повідомлень? Як буде здійснюватися блоковий код Хаффмана?
- 4.Як здійснюється кодування/декодування вхідних даних за адаптивним алгоритмом Хаффмана?
- 5.Які недоліки словниковых алгоритмів LZ77, LZSS?

Варіант 1

- 1.Що таке надлишковість коду? Як вона визначається?
- 2.Які переваги та недоліки використання оптимального кодування Шеннона-Фано і Хаффмана?
- 3.У чому полягає метод блокування повідомлень? Як буде здійснюватися блоковий код Хаффмана?
- 4.Як здійснюється кодування/декодування вхідних даних за адаптивним алгоритмом Хаффмана?
- 5.У чому полягають словникові методи стиснення LZ78, LZW? Чим визначається довжина кодів для цих алгоритмів? Які переваги модифікації LZW?

Варіант 1

- 1.Що таке економне кодування інформації? З якою метою воно здійснюється?
- 2.Яке кодування інформації називається статистичним? Які алгоритми стиснення даних відносять до статистичних?
- 3.Чим визначається верхня границя стиснення інформації? Які існують граници стиснення при використанні оптимального кодування Шеннона-Фано і Хаффмана?
- 4.У чому полягає арифметичний алгоритм кодування інформації? Які його переваги в порівнянні з іншими статистичними методами стиснення інформації?
- 5.У чому полягає основна ідея словниковых методів стиснення інформації? У чому переваги використання словниковых методів у порівнянні із статистичними?

Варіант 1

- 1.Які існують способи задання кодів?
- 2.Що таке оптимальне кодування інформації? Який критерій оптимальності статистичних кодів?
- 3.Чим визначається верхня границя стиснення інформації? Які існують граници стиснення при використанні оптимального кодування Шеннона-Фано і Хаффмана?
- 4.Як здійснюється декодування даних за арифметичним алгоритмом?
- 5.На які основні групи поділяються словникові алгоритми стиснення? Які їх відмітні риси?

Варіант 1

- 1.Що таке рівномірні й нерівномірні коди?
- 2.Які коди називаються префіксними? Що таке вектор Крафта? Як записується нерівність Крафта? У чому полягає умова оптимальності префіксних кодів?
- 3.Які коди характеризуються «наявністю пам'яті»?
- 4.Як здійснюється декодування даних за арифметичним алгоритмом?
- 5.У чому полягають словникові методи стиснення з використанням «ковзного» вікна LZ77, LZSS? Чим визначається довжина кодів цих алгоритмів?

Додаток В Приклади питань і варіантів завдань до залікової контрольної роботи

Питання до залікової контрольної роботи

1. Поняття теорії інформації, кодування. Галузі застосування, напрями.
2. Інформаційна ентропія.. Повідомлення.
3. Властивості ентропії. Вимоги до виміру і функції.
4. Інформаціона ентропія за основою g і 2 . Часткова ентропія.
5. Які існують види інформації?
6. Як перевести неперервну інформацію в дискретний (цифровий) вигляд?
7. Що таке частота дискретизації неперервної інформації?
8. Як формулюється теорема дискретизації?
9. Класифікація кодів.
10. Характеристика системи передавання даних.
11. Способи вимірювання інформації.
12. Інтуїтивні властивості інформації.
13. Кількість інформації взаємозалежних повідомлень
14. Ентропія джерела.
15. Властивості ентропії джерела. Формула Хартлі.
16. Надлишковість джерела.
17. Розподіл сумісних імовірностей десертних випадкових величин. Кількість інформації
18. Властивості кількості інформації і ентропії
19. Задача на знаходження ентропії дискретних випадкових величин.
20. Що таке економне кодування інформації? З якою метою воно здійснюється?
21. Які існують способи задання кодів?
22. Що таке рівномірні й нерівномірні коди?
23. З якою метою використовуються оптимальні нерівномірні коди?
24. Що таке надлишковість коду? Як вона визначається?
25. Яке кодування інформації називається статистичним? Які алгоритми стиснення даних відносять до статистичних?
26. Що таке оптимальне кодування інформації? Який критерій оптимальності статистичних кодів?
27. Які коди називаються префіксними? Що таке вектор Крафта? Як записується нерівність Крафта? У чому полягає умова оптимальності префіксних кодів?
28. У чому полягає алгоритм побудови оптимального коду Шеннона-Фано?
29. У чому полягає алгоритм побудови оптимального коду Хаффмана?
30. Які переваги та недоліки використання оптимального кодування Шеннона-Фано і Хаффмана?
31. Чим визначається верхня границя стиснення інформації? Які існують граници стиснення при використанні оптимального кодування Шеннона-Фано і Хаффмана?
32. Які коди характеризуються «наявністю пам'яті»?
33. Які коди називаються блоковими? Що таке порядок блокового коду?
34. У чому полягає метод блокування повідомлень? Як будується блоковий код Хаффмана?
35. У чому полягає арифметичний алгоритм кодування інформації? Які його переваги в порівнянні з іншими статистичними методами стиснення інформації?

36. Як здійснюється декодування даних за арифметичним алгоритмом?
37. У чому полягає адаптивний алгоритм Хаффмана? Що таке упорядковане дерево Хаффмана?
38. Як здійснюється кодування/декодування вхідних даних за адаптивним алгоритмом Хаффмана?
39. Які переваги і недоліки адаптивного алгоритму Хаффмана?
40. У чому полягає основна ідея словниковых методів стиснення інформації? У чому переваги використання словниковых методів у порівнянні із статистичними?
41. На які основні групи поділяються словникові алгоритми стиснення? Які їх відмітні риси?
42. У чому полягають словникові методи стиснення з використанням «ковзного» вікна LZ77, LZSS? Чим визначається довжина кодів цих алгоритмів?
43. Які переваги модифікованого алгоритму LZSS у порівнянні з LZ77?
44. Які недоліки словниковых алгоритмів LZ77, LZSS?
45. У чому полягають словникові методи стиснення LZ78, LZW? Чим визначається довжина кодів для цих алгоритмів? Які переваги модифікації LZW?
46. Які переваги алгоритмів LZ78, LZW у порівнянні з LZ77, LZSS?

Приклад залікової контрольної роботи

1. Класифікація кодів.
2. Що таке економне кодування інформації? З якою метою воно здійснюється?
3. З якою метою використовуються оптимальні нерівномірні коди?
4. Що таке оптимальне кодування інформації? Який критерій оптимальності статистичних кодів?
5. У чому полягає алгоритм побудови оптимального коду Хаффмана?
6. Які коди характеризуються «наявністю пам'яті»?
7. У чому полягає арифметичний алгоритм кодування інформації? Які його переваги в порівнянні з іншими статистичними методами стиснення інформації?
8. Як здійснюється кодування/декодування вхідних даних за адаптивним алгоритмом Хаффмана?
9. На які основні групи поділяються словникові алгоритми стиснення? Їх особливості ?
10. Які недоліки словниковых алгоритмів LZ77, LZSS?
11. Побудувати таблиці кодів і обчислити їх середню довжину за алгоритмами Шеннона-Фано, Хаффмана і арифметичним для дискретної випадкової величини X , заданої таким розподілом:

| X | 1 | 2 | 3 | 4 | 5 | 6 |
|----------|----------|----------|----------|----------|----------|----------|
| P | 0,4 | 0,2 | 0,05 | 0,05 | 0,2 | 0,1 |

12 Розкодувати повідомлення ‘A’0‘F’00‘X’01111101010110111101 00101, закодоване за адаптивним алгоритмом Хаффмана. Обчислити довжини кодів стиснутого і нестиснутого повідомлення в бітах.

13 Розкодувати повідомлення ‘D’0‘B’0100‘C’000‘A’11010 11111110, закодоване за адаптивним алгоритмом Хаффмана. Обчислити довжини кодів стиснутого і нестиснутого повідомлення у бітах.

14 Розглянути лінійний блоковий (k, n) -код, кодове слово якого утворюється за правилом $\mathbf{u}=(x_1, x_2, x_3, x_4, x_1+x_3+x_4, x_1+x_2+x_4, x_2+x_3+x_4)$. Визначити перевірну матрицю коду і параметри n , k . Чому дорівнює мінімальна кодова відстань отриманого коду?

15 Твірна матриця лінійного блокового коду має вигляд

$$\mathbf{G}_{(2,6)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Визначити мінімальну відстань Хеммінга даного коду і кратність помилок, що може виявити/ виправити код. Яка ймовірність невиявлення помилки кодом? Побудувати таблицю декодування. Виправити помилкові комбінації **101110**, **010100**, **101111**.

16 Ітеративний код заданий матрицею вигляду

$$\begin{pmatrix} m_0 & m_1 & p_0 \\ m_2 & m_3 & p_1 \\ p_2 & p_3 & p_4 \end{pmatrix}.$$

Записати твірну матрицю еквівалентного лінійного блокового коду. Визначити здатність коду виявляти/виправляти помилки, ймовірність невиявлення помилки, якщо ймовірність помилок у каналі $q=10^{-4}$.

17 Розпакувати повідомлення і обчислити довжину їх кодів. Повідомлення стиснute за алгоритмом **LZ78** (словник – 16 фраз). Код стиснутого повідомлення: **<0,’/’> <0,’W’> <0,’E’> <0,’D’> <1,’W’> <3,’/’> <2,’E’> <4,’/’> <7,’B’> <5,’E’> <6,’W’> <3,’B’> <10,’E’> <1,’E’>**

Додаток Г Приклади індивідуальних завдань

Завдання А

1. Знайти кількість взаємної інформації $I(X_1, X_2)$, якщо сумісний розподіл ймовірностей системи ДВВ X_1, X_2 такий:

| | | | | |
|-------|-------|-------|-------|-------|
| X_1 | 0 | 0 | 1 | 1 |
| X_2 | 0 | 1 | 0 | 1 |
| P | $1/6$ | $1/3$ | $1/4$ | $1/4$ |

2. ДВВ X_1 з однаковими ймовірностями може набувати значень **-1, 0, 1**, а ДВВ X_2 з однаковими ймовірностями – значень **0, 1**. X_1 та X_2 – незалежні ДВВ, $Y=X_1^2 - X_2^2$. Знайти кількість взаємної інформації $I(Y, X_1)$, $I(Y, X_2)$ та ентропії HX_1 , HX_2 , HY .
3. Знайти ентропії ДВВ X, Y, Z і кількість інформації, що містить ДВВ $Z=|X-Y| \bmod 3$ стосовно X та стосовно Y . X, Y – незалежні ДВВ, задані такими розподілами:

| | | | | |
|-----|-------|-------|-------|-------|
| X | 1 | 2 | 3 | 4 |
| P | $1/8$ | $1/8$ | $1/4$ | $1/2$ |

| | | | | |
|-----|-------|-------|-------|-------|
| Y | 1 | 2 | 3 | 4 |
| P | $1/8$ | $1/2$ | $1/4$ | $1/8$ |

4. Знайти ентропії ДВВ X, Y, Z і кількість інформації, що містить ДВВ $Z=X+Y$ стосовно ДВВ Y . X та Y незалежні ДВВ., задані такими розподілами:

| | | | | | | | |
|---|-------|-------|-------|-------|---|-------|-------|
| X | 0 | 1 | 3 | 4 | Y | -2 | 2 |
| P | $1/8$ | $1/8$ | $1/4$ | $1/2$ | P | $3/8$ | $5/8$ |

5. Скільки інформації про ДВВ X_1 містить ДВВ $Z=(X_1+1)^2-X_2$, якщо незалежні ДВВ X_1 та X_2 можуть із однаковою ймовірністю набувати значень 0 або 1? Знайти ентропії HX_1 , HZ . Який характер залежності між X_1 та Z ?

6. Знайти ентропії ДВВ X, Y, Z і кількість інформації, що містить ДВВ $Z=(X+Y)\bmod 2$ стосовно ДВВ Y . X та Y незалежні ДВВ., задані такими розподілами ймовірностей:

| | | | | | | | |
|---|------|------|------|------|---|------|------|
| X | 0 | 1 | 3 | 4 | Y | 2 | 4 |
| P | 0,35 | 0,27 | 0,13 | 0,25 | P | 0,76 | 0,24 |

7. ДВВ X_1, X_2 залежні й можуть із однаковою ймовірністю набувати значення **0** або **1**. Знайти кількість взаємної інформації $I(X_1, X_2)$, якщо сумісний розподіл ймовірностей системи ДВВ X_1, X_2 такий:

| | | | | |
|-------|----------|----------|----------|----------|
| X_1 | 0 | 0 | 1 | 1 |
| X_2 | 0 | 1 | 0 | 1 |
| P | 0,17 | 0,25 | 0,32 | 0,26 |

8. ДВВ X_1 з однаковими ймовірностями може набувати значень **-1, 0, 1**, а ДВВ X_2 з однаковими ймовірностями – значень **0, 1**. X_1 та X_2 – незалежні ДВВ, $Y=X_1^2 - X_2^2$. Знайти кількість взаємної інформації $I(Y, X_1)$, $I(Y, X_2)$ та ентропії HX_1 , HX_2 , HY .

9. Знайти ентропії ДВВ X , Y , Z і кількість інформації, що містить ДВВ $Z=|X-Y| \bmod 3$ стосовно X та стосовно Y . X , Y – незалежні ДВВ, задані такими розподілами:

| | | | | |
|-----|----------|----------|----------|----------|
| X | 1 | 2 | 3 | 4 |
| P | 0,12 | 0,19 | 0,31 | 0,38 |

| | | | | |
|-----|----------|----------|----------|----------|
| Y | 1 | 2 | 3 | 4 |
| P | 0,21 | 0,43 | 0,24 | 0,12 |

10. Скільки інформації про ДВВ X_1 містить ДВВ $Z=(X_1+1)^2-X_2$, якщо незалежні ДВВ X_1 та X_2 можуть із однаковою ймовірністю набувати значень 0 або 1? Знайти ентропії HX_1 , HZ . Який характер залежності між X_1 та Z ?

11. ДВВ X з різною ймовірністю може набувати значень від **1** до **8**. ДВВ Y набуває значення **0**, якщо X парне, і **1**, якщо X непарне. Знайти кількість інформації $I(Y, X)$ і ентропію HX , якщо ДВВ X задана таким розподілом ймовірностей:

| | | | | | | | | |
|-----|----------|----------|----------|----------|----------|----------|----------|----------|
| X | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| P | 0,11 | 0,19 | 0,06 | 0,09 | 0,08 | 0,07 | 0,3 | 0,1 |

12. Знайти ентропії ДВВ X , Y , Z і кількість інформації, що містить ДВВ $Z=X+Y$ стосовно ДВВ Y . X та Y незалежні ДВВ., задані такими розподілами:

| | | | | | | | |
|-----|------|------|------|------|-----|------|------|
| X | 0 | 1 | 3 | 4 | Y | -2 | 2 |
| P | 0,14 | 0,15 | 0,27 | 0,44 | P | 0,43 | 0,57 |

13. Знайти ентропії ДВВ X , Y , Z і кількість інформації, що містить ДВВ $Z=2X+Y$ стосовно X та Y . X , Y – незалежні ДВВ, задані такими розподілами ймовірностей:

| | | | |
|-----|-----------|----------|----------|
| X | -1 | 0 | 1 |
| P | 0,25 | 0,48 | 0,27 |

| | | | |
|-----|----------|----------|----------|
| Y | 0 | 1 | 2 |
| P | 0,18 | 0,66 | 0,16 |

14. Знайти ентропії ДВВ X , Y , Z і кількість інформації, що містить ДВВ $Z=X^*Y$ стосовно X та Y . X , Y – незалежні ДВВ, задані такими розподілами:

| | | | | | | | |
|-----|----------|----------|----------|----------|-----|-----------|----------|
| X | 0 | 1 | 3 | 4 | Y | -2 | 2 |
| P | 0,16 | 0,14 | 0,29 | 0,41 | P | 0,42 | 0,58 |

15. ДВВ X_1 з однаковими ймовірностями може набувати значень **-1, 0, 1**, а ДВВ X_2 з однаковими ймовірностями – значень **0, 1**. X_1 та X_2 – незалежні ДВВ, $Y=X_1^2+X_2^2$. Знайти кількість взаємної інформації $I(Y, X_1)$, $I(Y, X_2)$ та ентропії HX_1 , HX_2 , HY .

16. Існують двоє пошкоджених чотирьохроздядних кодових замків. Перший має заблокованими перший, другий і четвертий розряди 1, 0, 0 відповідно, а також заблоковані дві цифри 5 і 6 у третьому розряді. У другому замку у третьому і четвертому розрядах можуть одночасно бути встановлені парні або непарні цифри. Наприклад, в першому встановлена парна, то в другому теж тільки парна цифра. Який із замків має більшу ентропію.
17. Для систем повідомлень X, Y задано матрицю сумісних ймовірностей

$$P(X, Y) = \begin{pmatrix} 0,3 & 0 & 0 \\ 0,2 & 0,3 & 0,1 \\ 0 & 0,1 & 0 \end{pmatrix}$$

Визначити повні умовні ентропії $H(X/Y)$, $H(Y/X)$.

18. Сигнал з амплітудою A_1 (подія x_1) або з амплітудою A_2 (подія x_2) та зсувом фаз f_1 (подія y_1) або зсувом фаз f_2 (подія y_2) визначено ймовірностями сумісних подій $p(x_1, y_1) = 0,73$, $p(x_1, y_2) = 0,21$, $p(x_2, y_1) = 0,02$, $p(x_2, y_2) = 0,04$.

Обчислити ентропію об'єднаних залежних подій.

19. Джерелом повідомлень генеруються символи 0 та 1 з ймовірностями $p(0) = 3/4$, $p(1) = 1/4$. Умовні ймовірності для отриманих повідомлень $p(0/0) = 2/3$, $p(1/0) = 1/3$, $p(0/1) = 1$, $p(1/1) = 0$.

20. Обчислити ентропію джерела повідомлень за умови ймовірності появи символів на вході приймача становить $p(b1) = 0,1$, $p(b2) = 0,3$, $p(b3) = 0,4$, $p(b4) = 0,2$, а матриця переходів

$$P(a/b) = \begin{pmatrix} 0,99 & 0,02 & 0 & 0 \\ 0,01 & 0,98 & 0,01 & 0,01 \\ 0 & 0 & 0,98 & 0,02 \\ 0 & 0 & 0,01 & 0,97 \end{pmatrix}.$$

21. Обчислити ентропію джерела повідомлень за умови рівноймовірності появи символів джерела X, якщо матриця завад визначено матрицею

$$P(y/x) = \begin{pmatrix} 0,96 & 0,04 & 0 \\ 0,03 & 0,95 & 0,02 \\ 0,02 & 0,04 & 0,94 \end{pmatrix}.$$

22. Обчислити ентропію джерела повідомлень за умови ймовірності появи символів на вході приймача становить $p(x_1) = 0,5$, $p(x_2) = 0,3$, $p(x_3) = 0,2$, а матриця переходів має вигляд

$$P(y/x) = \begin{pmatrix} 0,97 & 0,03 & 0 \\ 0,01 & 0,98 & 0,01 \\ 0 & 0,04 & 0,96 \end{pmatrix}.$$

23. Для систем повідомлень X, Y задано матрицю сумісних ймовірностей

$$P(X, Y) = \begin{pmatrix} 0,3 & 0 & 0 \\ 0,2 & 0,3 & 0,1 \\ 0 & 0,1 & 0 \end{pmatrix}$$

Визначити повні умовні ентропії $H(X/Y), H(Y/X)$.

24. Матриця сумісних ймовірностей каналу зв'язку має вигляд

$$p(x_i, y_j) = \begin{bmatrix} 0,29 & 0,005 & 0,005 \\ 0,005 & 0,19 & 0,005 \\ 0,005 & 0,005 & 0,49 \end{bmatrix}.$$

Знайти часткові й загальну умовні ентропії $H(X/y_j), H(X/Y)$ і взаємну інформацію $I(X, Y)$.

25. Матриця сумісних ймовірностей каналу зв'язку має вигляд

$$p(x_i, y_j) = \begin{bmatrix} 0,29 & 0,005 & 0,005 \\ 0,005 & 0,19 & 0,005 \\ 0,005 & 0,005 & 0,49 \end{bmatrix}.$$

Знайти часткові й загальну умовні ентропії $H(X/y_j), H(X/Y)$ і взаємну інформацію $I(X, Y)$.

26. Канал передачі інформації заданий ансамблем $\{x_1, x_2, x_3\}$ з ймовірностями $\{0,3; 0,2; 0,5\}$. Матриця умовних ймовірностей каналу має вигляд

$$p(y_j / x_i) = \begin{bmatrix} 0,97 & 0,015 & 0,015 \\ 0,015 & 0,97 & 0,015 \\ 0,015 & 0,015 & 0,97 \end{bmatrix}.$$

Знайти інформаційні втрати в каналі, пропускну здатність каналу й швидкість передачі повідомлень джерелом, якщо час передачі одного повідомлення $\tau=10^{-3} \text{ c}$.

27. Знайти кількість переданої інформації в одному повідомленні джерела й пропускну здатність каналу зв'язку при $\tau=10^{-3} \text{ c}$, де τ - час, затрачуваний на передачу одного повідомлення, якщо матриця сумісних ймовірностей каналу має вигляд

$$p(x_i, y_j) = \begin{bmatrix} 0,2 & 0 & 0,1 \\ 0 & 0,2 & 0,1 \\ 0,1 & 0,1 & 0,2 \end{bmatrix}.$$

28. Знайти інформаційні втрати в каналі й пропускну здатність дискретного каналу зв'язку при $\tau=10^{-3} \text{ c}$, де τ - час, затрачуваний на передачу одного повідомлення, якщо матриця сумісних ймовірностей каналу має вигляд

$$p(x_i, y_j) = \begin{bmatrix} 0,63 & 0,035 & 0,035 \\ 0,02 & 0,16 & 0,02 \\ 0,017 & 0,013 & 0,07 \end{bmatrix}.$$

Чи можлива безпомилкова передача інформації в цьому каналі, якщо продуктивність джерела $V_{дж}=860 \text{ сим/с}$?

29. Дослідження каналу зв'язку між джерелом X і спостерігачем Y виявило такі умовні ймовірності вибору повідомлень $y_j \in Y$:

$$p(y_j / x_i) = \begin{bmatrix} 0,9 & 0,05 & 0,05 \\ 0,1 & 0,8 & 0,1 \\ 0,17 & 0,13 & 0,7 \end{bmatrix}.$$

Знайти часткові й загальну умовні ентропії повідомлень у цьому каналі $H(X/y_j)$, $H(X/Y)$, якщо джерело задане ансамблем $\{x_1, x_2, x_3\}$ з ймовірностями $\{0,7; 0,2; 0,1\}$.

30. Джерело повідомлень X задано ансамблем $\{x_1, x_2, x_3\}$ з ймовірностями $p(x_1)=0,65$, $p(x_2)=0,25$, $p(x_3)=0,1$. Матриця умовних ймовірностей каналу має вигляд

$$p(y_j / x_i) = \begin{bmatrix} 0,99 & 0,005 & 0,005 \\ 0,13 & 0,75 & 0,12 \\ 0,15 & 0,35 & 0,5 \end{bmatrix}.$$

Знайти кількість інформації, передану в одному з 100 повідомленнях джерела, інформаційні втрати в каналі при передачі 100 повідомлень з алфавіту X ?

Завдання В

1. Для заданого **коду** і **виразу** за індивідуальним завданням скласти таблицю розподілу ймовірностей букв(блоків) алфавіту джерела.
2. Обчислити ентропію джерела.
3. Обчислити надлишковість джерела
4. Побудувати код джерела алфавіту.
5. Обчислити середню довжину коду
6. Обчислити надлишковість отриманого коду..
7. Закодувати заданий вираз отриманим кодом
8. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах і закодованого виразу.
9. Оформити звіт з виконання етапів 1-8.

| Варіант | Завдання | Вираз | Код |
|----------------|-----------------|---|----------------------------------|
| 1. | 49 | Науки ні вода не затопить, ні огонь не спалить. | Код Хаффмена |
| 2. | 48 | Незнайко на печі лежить, а знайко по дорозі біжить. | Код Хаффмена другого порядку |
| 3. | 47 | Не бажай синові багатства, а бажай розуму! | Код Хаффмена |
| 4. | 46 | Не краса красить, а розум. | Код Хаффмена |
| 5. | 45 | Не лінися рано вставати та змолоду більше знати! | Код Хаффмена другого порядку |
| 6. | 44 | Не на користь книжку читать, коли вершки лише хапать. | Код Шеннона-Фано другого порядку |
| 7. | 43 | Не перо пише, а розум. | Код Хаффмена |
| 8. | 42 | Один розум добре, а два ще краще! | Код Шеннона-Фано |
| 9. | 41 | Одна розумна голова добре, а дві ще краще! | Код Шеннона-Фано |
| 10. | 40 | Око бачить далеко, а розум ще дальше. | Код Хаффмена |
| 11. | 39 | Перемагай труднощі розумом, а небезпеку — досвідом! | Код Хаффмена третього порядку |
| 12. | 38 | Треба розумом надточiti, де сила не візьме. | Код Шеннона-Фано другого порядку |
| 13. | 37 | У сусіда ума не позичиш. | Код Хаффмена |
| 14. | 36 | Учений іде, а неук слідом спотикається. | Код Хаффмена другого порядку |
| 15. | 35 | Учись змолоду — пригодиться на старість! | Код Шеннона-Фано |
| 16. | 34 | Учись — на старість буде як нахідка! | Код Хаффмена |
| 17. | 33 | Розумний всякому дає лад. | Код Хаффмена |
| 18. | 32 | Розумну річ приемно й слухать. | Код Шеннона-Фано |
| 19. | 31 | Розум — скарб людини. | Код Хаффмена |
| 20. | 50 | Сила без голови шаліє, а розум без сили мліє. | Код Шеннона-Фано другого порядку |
| 21. | 51 | Сила та розум — краса людини. | Код Хаффмена |

| | | | |
|-----|----|--|------------------------------|
| 22. | 52 | Сила уму уступає! | Код Хаффмена |
| 23. | 53 | Скільки голів, стільки й умів! | Код Хаффмена |
| 24. | 54 | Старого горобця на половині не обдуриш. | Код Хаффмена |
| 25. | 55 | Хто вчиться змолоду, не зазнає на старість голоду. | Код Хаффмена другого порядку |
| 26. | 56 | Хто грамоти вміє, той краще сіє. | Код Хаффмена |
| 27. | 57 | Хто добре учиться, той буде й добре робить. | Код Хаффмена другого порядку |
| 28. | 58 | Хто знання має, той мур зламає. | Код Хаффмена |
| 29. | 59 | Хто людей питає, той і розум має. | Код Хаффмена |
| 30. | 60 | Хто хоче багато знати, тому треба мало спати. | Код Хаффмена другого порядку |

Завдання В

10. Для заданого **коду** і **виразу** за індивідуальним завданням скласти таблицю розподілу ймовірностей букв(блоків) алфавіту джерела.
11. Обчислити ентропію джерела.
12. Обчислити надлишковість джерела
13. Побудувати код джерела алфавіту.
14. Обчислити середню довжину коду
15. Обчислити надлишковість отриманого коду..
16. Закодувати заданий вираз отриманим кодом
17. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах і закодованого виразу.
18. Оформити звіт з виконання етапів 1-8.

| Варіант | Завдання | Вираз | Код |
|----------------|-----------------|---|----------------------------------|
| 1. | 49 | Щастя без розуму — торбина дірява. | Код Хаффмена |
| 2. | 48 | Що в молодості навчишся, то на старість як знайдеш. | Код Хаффмена другого порядку |
| 3. | 47 | Що голова, то й розум! | Код Хаффмена |
| 4. | 46 | Що знаєш, що вмієш, те за плечима не носить. | Код Шеннона-Фано другого порядку |
| 5. | 45 | Щоб часом дарма не блудить, чужого розуму питайся. | Код Хаффмена другого порядку |
| 6. | 44 | Як голова сивіє, то чоловік мудріє. | Код Шеннона-Фано другого порядку |
| 7. | 43 | Яка головонька, така й розмовонька! | Код Хаффмена |
| 8. | 42 | Які літа — такий розум! | Код Шеннона-Фано |
| 9. | 41 | З віком розум приходить. | Код Хаффмена |
| 10. | 40 | Як жили наші діди та прадіди, так нам веліли. | Код Хаффмена другого порядку |
| 11. | 39 | Де є життя, там є й надія. | Код Хаффмена |
| 12. | 38 | Життя біжить — як музика дзвенить. | Код Шеннона-Фано другого порядку |
| 13. | 37 | Які люди, таке і життя буде. | Код Хаффмена |
| 14. | 36 | Живий про живе гадає. | Код Хаффмена другого порядку |
| 15. | 35 | Був на коні і під конем. | Код Шеннона-Фано |
| 16. | 34 | I коза сита, і капуста ціла. | Код Хаффмена |
| 17. | 33 | Був би послом, якби не вдався ослом. | Код Хаффмена другого порядку |
| 18. | 32 | Собака бреше, а кінь іде. | Код Шеннона-Фано |
| 19. | 31 | I ведмедя вчать танцювати. | Код Хаффмена |
| 20. | 50 | Дружба з вовком виласить боком. | Код Шеннона-Фано другого порядку |
| 21. | 51 | Ми про вовка, а вовк суне. | Код Хаффмена |
| 22. | 52 | За вовка помовка, а вовк іде. | Код Хаффмена |
| 23. | 53 | Лев - цар звірів, а боїться комарів. | Код Хаффмена |
| 24. | 54 | Не бий бобра — не буде добра. | Код Хаффмена |
| 25. | 55 | Що за холод, як козак молод. | Код Хаффмена |

| | | | |
|-----|----|---|------------------------------|
| 26. | 56 | Семеро воює, а один горює. | Код Хаффмена |
| 27. | 57 | Де дружніші, там і сильніші | Код Хаффмена |
| 28. | 58 | Здобудеш освіту – побачиш більше світу. | Код Хаффмена другого порядку |
| 29. | 59 | Терпіння дає уміння. | Код Хаффмена |
| 30. | 60 | Дурний друг — недруг. | Код Хаффмена |

Завдання С

1. Для заданого **коду і виразу** за алгоритмом індивідуального завдання побудувати таблицю кодування (закодувати) повідомлення виразу. Для арифметичного алгоритму визначити таблицю розподілу символів.
2. Обчислити ентропію джерела.
3. Виконати декодування закодованого за пунктом 1 повідомлення виразу.
4. Побудувати дерево кодування **виразу** індивідуального завдання кодом Хаффмена.
5. Закодувати заданий **вираз** отриманим кодом Хаффмена
6. Обчислити довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим **кодом** (алгоритмом) і . кодом Хаффмена
7. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим кодом (алгоритмом) і . кодом Хаффмена. Зробити висновки
8. Оформити звіт з виконання етапів 1-7.

| Варіант | Завдання | Вираз | Переклад | Код |
|---------|----------|------------------------|--|------------------------------|
| 1. | 90 | Sapienti sat | Розумному досить. | Адаптивний алгоритм Хаффмена |
| 2. | 89 | Scientia potentia est | Знання - сила. | Арифметичний алгоритм |
| 3. | 88 | Si sapis, sis apis | Якщо ти розумний, будь (працьовитим), як бджола. | Арифметичний алгоритм |
| 4. | 87 | Sine ira et studio | Без гніву і пристрасті. | Адаптивний алгоритм Хаффмена |
| 5. | 86 | Sol lucet omnibus | Сонце світить всім. | Арифметичний алгоритм |
| 6. | 85 | Spes vana | Марна надія. | Адаптивний алгоритм Хаффмена |
| 7. | 84 | Sui generis | Свого роду, своєрідний. | Арифметичний алгоритм |
| 8. | 83 | Suum cuique | Кожному своє. | Адаптивний алгоритм Хаффмена |
| 9. | 82 | Tabula rasa | Чиста дошка; щось неторкнуте. | Адаптивний алгоритм Хаффмена |
| 10. | 81 | Tempori pace! | Бережи час! | Арифметичний алгоритм |
| 11. | 80 | Terra incognita | Невідома земля; щось незнане. | Арифметичний алгоритм |
| 12. | 61 | Tertium non datur | Третього не дано. | Адаптивний алгоритм Хаффмена |
| 13. | 62 | Theoria cum praxi | Теорія з практикою. | Арифметичний алгоритм |
| 14. | 63 | Ubi mel, ibi fel | Де мед, там і жовч. | Адаптивний алгоритм Хаффмена |
| 15. | 64 | Ultima ratio | Останній довід. | Арифметичний алгоритм |
| 16. | 65 | Veni, vidi, vici | Прийшов, побачив, переміг. | Арифметичний алгоритм |
| 17. | 66 | Verba magistri | Слова вчителя, авторитетної людини. | Арифметичний алгоритм |
| 18. | 67 | Viribus unitis | Спільними зусиллями. | Адаптивний алгоритм Хаффмена |
| 19. | 68 | Vivere est cogitare | Жити - означає мислити. | Арифметичний алгоритм |
| 20. | 69 | Vivere est militare | Жити - означає боротися. | Адаптивний алгоритм Хаффмена |
| 21. | 70 | Alea iacta est | Жереб кинуто | Арифметичний алгоритм |
| 22. | 71 | Amat victoria curam | Перемога любить старанність | Арифметичний алгоритм |
| 23. | 72 | Ars longa, vita brevis | Мистецтво вічне, життя коротке | Адаптивний алгоритм Хаффмена |
| 24. | 73 | Aurea mediocritas | Золота середина | Арифметичний алгоритм |
| 25. | 74 | Aut Caesar, aut nihil | Або Цезар, або ніхто | Адаптивний алгоритм Хаффмена |

| | | | | |
|-----|----|---------------------------|---|---------------------------------|
| 26. | 75 | Labores pariunt honores | Праця породжує пошану | Адаптивний алгоритм Хаффмена |
| 27. | 76 | Liber est mutus magister | Книга - це німий учитель | Адаптивний алгоритм Хаффмена |
| 28. | 79 | Mens sana in corpore sano | У здоровому тілі здоровий дух | Адаптивний алгоритм Хаффмена |
| 29. | 78 | Nil desperandum | Не впадай у відчай. | Арифметичний алгори |
| 30. | 77 | Non multa, sed multum | Не багато (за кількістю), але багато (за значенням). | Адаптивний алгоритм Хаффмена |

Завдання С

9. Для заданого **коду** і **виразу** за алгоритмом індивідуального завдання побудувати таблицю кодування (закодувати) повідомлення виразу. Для арифметичного алгоритму визначити таблицю розподілу символів.
10. Обчислити ентропію джерела.
11. Виконати декодування закодованого за пунктом 1 повідомлення виразу.
12. Побудувати дерево кодування **виразу** індивідуального завдання кодом Хаффмена.
13. Закодувати заданий **вираз** отриманим кодом Хаффмена
14. Обчислити довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим **кодом** (алгоритмом) і . кодом Хаффмена
15. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим кодом (алгоритмом) і . кодом Хаффмена. Зробити висновки
16. Оформити звіт з виконання етапів 1-7.

| Варіант | Завдання | Вираз | Переклад | Код |
|---------|----------|----------------------------|--|------------------------------|
| 1. | 90 | Cogito, ergo sum | Я мислю, отже, я існую. | Арифметичний алгоритм |
| 2. | 89 | Contra spem spero | Без надії сподіваюсь. | Адаптивний алгоритм Хаффмена |
| 3. | 88 | Cum grano salis | З дрібкою солі (в'їдливо). | Арифметичний алгоритм |
| 4. | 87 | Dictum – factum | Сказано - зроблено. | Адаптивний алгоритм Хаффмена |
| 5. | 86 | Docendo discimus | Навчаючи, вчимося. | Арифметичний алгоритм |
| 6. | 85 | Errare humanum est | Людині властиво помиллятися. | Адаптивний алгоритм Хаффмена |
| 7. | 84 | Eruditio aspera optima est | Суворе навчання - найкраще. | Арифметичний алгоритм |
| 8. | 83 | Est modus in rebus | Є міра в речах. | Адаптивний алгоритм Хаффмена |
| 9. | 82 | Et cetera | I так далі. | Адаптивний алгоритм Хаффмена |
| 10. | 81 | Finis coronat opus | Кінець вінчає діло. | Адаптивний алгоритм Хаффмена |
| 11. | 80 | Fortuna caeca est | Доля сліпа. | Арифметичний алгоритм |
| 12. | 61 | Debes, ergo potes | Ти повинен, отже, ти можеш. | Адаптивний алгоритм Хаффмена |
| 13. | 62 | Labor corpus firmat | Праця зміщює тіло. | Арифметичний алгоритм |
| 14. | 63 | Labor omnia vincit | Праця все перемагає | Адаптивний алгоритм Хаффмена |
| 15. | 64 | Natura non facit saltus | Природа не робить стрибків | Арифметичний алгоритм |
| 16. | 65 | Nota bene! (NB!) | Познач добре, зверни увагу | Арифметичний алгоритм |
| 17. | 66 | Nulla dies sine linea | Жодного дня без рядка (занять, праці). | Арифметичний алгоритм |
| 18. | 67 | Otia dant negotia | Лінощі породжують вади. | Адаптивний алгоритм Хаффмена |

| | | | | |
|-----|----|--------------------------|---|------------------------------|
| 19. | 68 | Otium post negotium | Відпочинок після роботи. | Арифметичний алгоритм |
| 20. | 69 | Fac et spera! | Роби і надійся! | Адаптивний алгоритм Хаффмена |
| 21. | 70 | Habent sua fata libelli | Книги мають свою долю. | Арифметичний алгоритм |
| 22. | 71 | Per aspera ad astra | Через терни до зірок. | Арифметичний алгоритм |
| 23. | 72 | Perfice te! | Удосконалуй себе! | Адаптивний алгоритм Хаффмена |
| 24. | 73 | Post factum | Після того, що сталося; заднім числом. | Арифметичний алгоритм |
| 25. | 74 | Post scriptum (P.S.) | Після написаного, приписка в кінці листа, постскріптум. | Адаптивний алгоритм Хаффмена |
| 26. | 75 | Dum spiro, spero | Доки дихаю, надіюсь | Арифметичний алгоритм |
| 27. | 76 | Potius sero quam nunquam | Краще пізніше, ніж ніколи. | Адаптивний алгоритм Хаффмена |

Завдання D

17. Кодувати окрім **вирази 1 і 2** за алгоритмом, довжинами словника s, буфера b, кількістю фраз f.
18. Обчислити довжини закодованих повідомлень і виконати їх порівняння.
19. Визначити коефіцієнти стискання виразів
20. Виконати декодування повідомлень виразів.
21. Оформити звіт з виконання етапів 1-4.

| Варіант | Завдання | Вираз 1 | Вираз 2 | Алгоритм |
|---------|----------|--|---|----------------|
| 28. | 99 | A bad compromise is better than a good lawsuit. | Поганий мир кращий доброї сварки. | LZ77 s=9, b=4 |
| 29. | 98 | A bad workman always blames his tools. | У поганого майстра погана пила. | LZ78 f=400 |
| 30. | 102 | A chain is no stronger than its weakest link. | Де тонко, там і рветься. | LZSS s=12, b=6 |
| 31. | 108 | A friend in need is a friend indeed. | Друг пізнається у біді. | LZW f=300 |
| 32. | 116 | A gift in the hand is better than two promises. | Краще синиця в жмені, ніж журавель в небі. | LZ77 s=10, b=5 |
| 33. | 103 | A man is known by the company he keeps. | Скажи мені, хто твій друг, і я скажу, хто ти. | LZ78 f=500 |
| 34. | 93 | A sound mind in a sound body. | В здоровому тілі здоровий дух. | LZSS s=8, b=5 |
| 35. | 92 | Appetite comes with the eating. | Апетит приходить під час їди. | LZW f=400 |
| 36. | 91 | Be in the wrong box. | Бути не в своїй тарілці | LZ77 s=13, b=6 |
| 37. | 117 | Before you know where you are. | I охнути не встигнеш. | LZ78 f=300 |
| 38. | 104 | Better be born lucky than rich. | Не родися красивим, а родися щасливим. | LZ77 s=12, b=6 |
| 39. | 109 | Better be the head of a dog, than the tail of a lion. | Краще бути першим серед останніх, ніж останнім серед перших | LZ78 f=400 |
| 40. | 115 | Better die standing than live kneeling. | Краще вмерти стоячи, ніж жити на колінах. | LZSS s=10, b=5 |
| 41. | 118 | Better late than never. | Краще пізно, ніж ніколи. | LZW f=500 |
| 42. | 110 | Books and friends should be few but good. | Істинні друзі, як і мудрі книги - рідкість. | LZ77 s=9, b=4 |
| 43. | 101 | Den ersten Tag ein Gast, den zweiten eine Last, den dritten stinkt er fast | Перший день гість-гість, другий день - тягар, третій день - майже ворог - | LZ78 f=600 |
| 44. | 100 | Der Alten Rat, der Jungen Tat macht Krummes grad | Рада старих і справи молодих розпрямляють криве | LZSS s=11, b=5 |
| 45. | 105 | Erst die Arbeit, dann das Spiel, nach der Reise kommt das Ziel | Спочатку робота, потім гра, мета досягається наприкінці подорожі | LZW f=500 |
| 46. | 114 | Was du heute kannst besorgen, | Не відкладай на завтра те, що | LZ77 |

| | | | | |
|-----|-----|---|---|------------------|
| | | das verschiebe nicht auf morgen | можеш зробити сьогодні | s=10,b=5 |
| 47. | 120 | Hitzig ist nicht witzig | Запальний - не розумний | LZ78 f=100 |
| 48. | 97 | Mitgegangen, mitgefangen, mitgehangen | Ми разом ходили, нас усіх разом піймали, всі разом були повішенні | LZSS s=13,b=6 |
| 49. | 96 | Ost und West, daheim das Best | Чи схід, чи захід, а будинку краще | LZW f=500 |
| 50. | 95 | Geteilte Freude ist doppelte Freude, geteilter Schmerz ist halber Schmerz | Розділена радість - подвійна радість, розділене горе - полгоря | LZ77 s=12,b=5 |
| 51. | 94 | August ohne Feuer macht das Brot teuer | Август без тепла - хліб втридорога | LZ78 f=400 |
| 52. | 119 | Darnach der Mann geraten, wird ihm die Wurst gebraten | Який людина, таку йому й ковбасу | LZSS s=15,b=6 |
| 53. | 106 | Heute stark, morgen im Sarg | Сьогодні сильний, завтра в труні | LZ77 s=9,b=4 |
| 54. | 113 | Im Dunkeln ist gut munkeln | У темряві легко говорити тайкома | LZ78 f=100 |
| 55. | 112 | Kein Vorteil ohne Nachteil | Немає переваг без недоліків | LZSS s=12,b=7 |
| 56. | 107 | Wer vieles beginnt, gar wenig zustande bringt | Хто багато починає, дуже мало здійснює | LZW f=500 |
| 57. | 111 | Wie der Hall, so der Stall | Яке луна, така стайня | LZ77 s=11,b=7 |

Завдання D

22. Кодувати окрім **вирази 1 і 2** за алгоритмом, довжинами словника s, буфера b, кількістю фраз f.
23. Обчислити довжини закодованих повідомлень і виконати їх порівняння.
24. Визначити коефіцієнти стискання виразів
25. Виконати декодування повідомлень виразів.
26. Оформити звіт з виконання етапів 1-4.

| Варіант | Завдання | Вираз 1 | Вираз 2 | Алгоритм |
|---------|----------|---|--|---------------|
| 31. | 99 | Call a spade a spade. | Називай речі своїми іменами. | LZ77s=9,b=4 |
| 32. | 98 | Danger foreseen is half avoided. | Якби знов, де впадеш, соломинки б підстелив. | LZ78 f=400 |
| 33. | 102 | Don't look a gift horse in the mouth. | Дарованому коневі в зуби не заглядають. | LZSS s=12,b=6 |
| 34. | 108 | Don't swap horses when crossing a stream. | Коней на переправі не міняють. | LZW f=300 |
| 35. | 116 | Don't trouble your trouble until your trouble troubles you. | Не буди лиха, поки лихо спить. | LZ77 s=10,b=5 |
| 36. | 103 | Dry bread at home is better than roast meat abroad. | В гостях добре, а вдома - краще. | LZ78 f=500 |
| 37. | 93 | Early to bed and early to rise, makes a man healthy, wealthy, and wise. | Хто рано лягає і рано встає, той здоров'я, багатство і розум наживе. | LZSS s=8,b=5 |
| 38. | 92 | Empty vessels make the greatest sound. | Пуста бочка найбільше скрипить. | LZW f=400 |
| 39. | 91 | Everything is good in its season. | На все свій час. | LZ77 s=13,b=6 |
| 40. | 117 | Evil communications corrupt good manners. | З вовками жити, по-вовчому вити. | LZ78 f=300 |
| 41. | 104 | Fear has magnifying eyes. | У страху великі очі. | LZ77 s=12,b=6 |
| 42. | 109 | Fish begins to stink at the head. | Риба гниє з голови. | LZ78 f=400 |
| 43. | 115 | Fools rush in where angels fear to tread. | Дурням закон не писаний. | LZSS s=10,b=5 |
| 44. | 118 | From tile sublime to the ridiculous is but a step. | Від великого до смішного - один крок. | LZW f=500 |
| 45. | 110 | Good health is above wealth. | Здоров'я важливіше від грошей. | LZ77 s=9,b=4 |
| 46. | 101 | Hawk will not pick out hawks' eyes. | Ворону ока не виклює. | LZ78 f=600 |
| 47. | 100 | He laughs best who laughs last. | Добре сміється той, хто сміється останнім. | LZSS s=11,b=5 |
| 48. | 105 | He that fears death lives not. | Двом смертям не бувати, а одної не минути. | LZW f=500 |
| 49. | 114 | He that would have eggs must | Любиш кататися - люби і | LZ77 |

| | | | | |
|-----|-----|---|---|------------------|
| | | endure the cackling of hens. | саночки возити. | s=10,b=5 |
| 50. | 120 | He who pleased everybody died before he was born. | Всім не вгодиш. | LZ78 f=100 |
| 51. | 97 | Der Kranke und der Gesunde haben ungleiche Stunde | У хворого й здорового час різне | LZSS s=13,b=6 |
| 52. | 96 | Ruh' und Rast ist halbe Mast | Відпочинок і спокій - половина годівлі | LZW f=500 |
| 53. | 95 | Leihen macht Freundschaft, Mahnen Feindschaft | Позичати - скріплювати дружбу, нагадувати - викликати ворожнечу | LZ77 s=12,b=5 |
| 54. | 94 | Was man spart vom Mund', fressen Katz' und Hund | Те, що відривають від свого рота, зжеруть кішка й собака | LZ78 f=400 |
| 55. | 119 | Unverhofft kommt oft | Несподіване приходить часто | LZSS s=15,b=6 |
| 56. | 106 | Ein Mann macht keinen Tanz, eine Blume keinen Kranz | Одна людина не складе танцю, з однієї квітки не зробиш вінка | LZ77 s=9,b=4 |
| 57. | 113 | Nicht jedes Holz gibt einen Bolz | Не із усякого дерева можна зробити стрілу | LZ78 f=100 |
| 58. | 112 | Kein Haus ohne Maus | Немає будинку без мишей | LZSS s=12,b=7 |
| 59. | 107 | In der Not schmeckt jedes Brot | У лиху всякий хліб смачний | LZW f=500 |
| 60. | 111 | Call a spade a spade. | Називай речі своїми іменами. | LZ77 s=11,b=7 |

**Додаток Д Приклади оформлення титульної сторінки виконання
індивідуального завдання**

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Практична робота №_____
з курсу «Теорія інформації і кодування»
Тема: ОСНОВИ ТЕОРІЇ ІНФОРМАЦІЇ ТА КОДУВАННЯ
Варіант ____

Виконав: студент курсу
групи

Прийняв:

Київ – 201_p.

Додаток Ж Приклади оформлення виконання індивідуального завдання

Задача 1

1. Постановка задачі

Джерело повідомлень X задано ансамблем $\{x_1, x_2, x_3\}$ з ймовірностями $p(x_1)=0,65; p(x_2)=0,25; p(x_3)=0,1$. Матриця умовних ймовірностей каналу має вигляд

$$p(y_j/x_i) = \begin{pmatrix} 0,99 & 0,005 & 0,005 \\ 0,13 & 0,750 & 0,120 \\ 0,15 & 0,350 & 0,500 \end{pmatrix}.$$

Знайти кількість інформації, передану в одному та 100 повідомленнях джерела, інформаційні втрати в каналі при передачі 100 повідомлень з алфавіту X ?

2. Математичний розв'язок задачі

Умовні ймовірності $p(y_j/x_i)$ при $i=j$ характеризують вплив завад у каналі зв'язку.

Часткова умовна ентропія спостерігача Y щодо джерела X визначається

$$H(Y/x_i) = - \sum_i p(y_j/x_i) * \log_2 p(y_j/x_i)$$

за формулою

Знайдемо часткові умовні ентропії для $i=1, \dots, 3$

$$H(Y/x_1) = -(0,99 * \log_2 0,99 + 0,005 * \log_2 0,005 + 0,005 * \log_2 0,005) = -(0,01435 - 0,382 - 0,382) = -(-0,0908) = 0,0908 \text{ (біт/сим)}.$$

$$H(Y/x_2) = -(0,13 * \log_2 0,13 + 0,75 * \log_2 0,75 + 0,12 * \log_2 0,12) = -(-0,38264 - 0,31129 - 0,36707) = 1,061 \text{ (біт/сим)}.$$

$$H(Y/x_3) = -(0,15 * \log_2 0,15 + 0,35 * \log_2 0,35 + 0,5 * \log_2 0,5) = -(-0,410545 - 0,5301 - 0,5) = 1,44065 \text{ (біт/сим)}$$

Інформаційні втрати визначаються умовною ентропією. Знаходимо загальну умовну ентропію ДВВ Y стосовно ДВВ X (для одного повідомлення):

$$H(Y/X) = \sum_i p(x_i) * H(Y/x_i) = 0,65 * 0,0908 + 0,25 * 1,061 + 0,1 * 1,44065 = 0,46834 \text{ (біт/сим)}.$$

$$H_{100}(Y/X) = 100 * 0,4683 = 43,83 \text{ (біт/сим)}$$

Скориставшись формулою множення ймовірностей

$$p(x_i, y_j) = p(x_i) * p(y_j/x_i),$$

побудуємо матрицю ймовірностей системи ДВВ \mathbf{X}, \mathbf{Y} :

$$p(x,y) = \begin{pmatrix} 0,99*0,65 & 0,005*0,65 & 0,005*0,65 \\ 0,13*0,25 & 0,75*0,25 & 0,12*0,25 \\ 0,15*0,10 & 0,35*0,10 & 0,50*0,10 \end{pmatrix} = \begin{pmatrix} 0,6435 & 0,00325 & 0,00325 \\ 0,0325 & 0,18750 & 0,03000 \\ 0,0150 & 0,03500 & 0,05000 \end{pmatrix}$$

Знайдемо $\mathbf{H}(Y)$. Для цього знайдемо $p(y_1), p(y_2), p(y_3)$, виконавши в матриці сумісних ймовірностей згортку за \mathbf{i} .

$$p(y_1) = 0,6435 + 0,0325 + 0,015 = 0,691$$

$$p(y_2) = 0,00325 + 0,1875 + 0,035 = 0,22575$$

$$p(y_3) = 0,00325 + 0,03 + 0,05 = 0,08325$$

Перевіряємо умову нормування:

$$p(y_1) + p(y_2) + p(y_3) = 0,691 + 0,22575 + 0,0825 = 1$$

Ентропія \mathbf{Y} :

$$\mathbf{H}(Y) = -(0,691 * \log_2 0,691 + 0,22575 * \log_2 0,22575 + 0,0825 * \log_2 0,0825) = -0,3685 - 0,484731 - 0,296956 = 1,1502$$

Перевіримо умову нормування

$$\sum_{ij} p_{ij} = 0,6435 + 0,00325 + 0,00325 + 0,0325 + 0,1875 + 0,03 + 0,015 + 0,035 + 0,05 = 1$$

Отже кількість інформації, передана в 1 повідомленні дорівнює

$$I(X, Y) = H(Y) - H(Y/X);$$

$$I(X, Y) = 1,1502 - 0,46834 = 0,68186 \text{ (біт/сим);}$$

$$\text{Для 100 повідомлень } I_{100}(X, Y) = 100 * 0,68186 = 68,186 \text{ (біт/сим)}$$

3. Відповідь

1)Інформаційні втрати, передані в 1 повідомленні

$$H(Y/X) = \sum_i p(x_i) * H(Y/x_i) = 0,46834 \text{ (біт/сим)}$$

2)Інформаційні втрати, передані в 100 повідомленнях

$$H_{100}(Y/X) = 100 * 0,46834 = 43,83 \text{ (біт/сим)}$$

3)Кількість інформації, передана в 1 повідомленні

$$I(X, Y) = 1,1502 - 0,46834 = 0,68186 \text{ (біт/сим)}$$

4)Кількість інформації, передана в 100 повідомленнях

$$I_{100}(X, Y) = 100 * 0,68186 = 68,186 \text{ (біт/сим)}$$

Задача 2

Постановка задачі

Варіант №4

Знайти ентропії ДВВ X, Y, Z і кількість інформації, що містить ДВВ $Z=X+Y$ стосовно Y . X, Y – незалежні ДВВ, задані такими розподілами:

Розподіл ймовірностей ДВВ X :

| | | | | |
|---|-----|-----|-----|-----|
| x | 0 | 1 | 3 | 4 |
| p | 1/8 | 1/8 | 1/4 | 1/2 |

Розподіл ймовірностей ДВВ Y :

| | | |
|---|-----|-----|
| y | -2 | 2 |
| q | 3/8 | 5/8 |

1. Загальні формули

Для розв'язку задачі використовуємо такі формули:

1. Ентропія джерела: $H(X) = - \sum_{i=1}^k p_i \log_2 p_i$, $i=1, \dots, k$, $i=1 \dots k$,

де k - об'єм алфавіту джерела для незалежних повідомлень.

2. Кількість інформації: $I(X, Y) = \sum_{i,j} p_{ij} \log_2 \frac{p_{ij}}{p_i q_j}$, де p_i, q_j – ймовірності, що

визначають закони розподілу ймовірностей ДВВ $P(X=X_i)=p_i, P(Y=Y_j)=q_j$;

p_{ij} - ймовірність, що визначає розподіл сумісних ймовірностей системи

$P(X=X_i, Y=Y_j) = p_{ij}$.

3. Розв'язок задачі

Скориставшись відповідним рядом розподілу ймовірностей ДВВ X та Y , знаходимо їх ентропії.

Ентропія ДВВ X :

$$H(X) = -$$

$$2 \cdot \frac{1}{8} \log_2 \frac{1}{8} + \frac{1}{4} \log_2 \frac{1}{4} + \frac{1}{2} \log_2 \frac{1}{2} = - \left(2 * \frac{1}{8} * (-3) + \frac{1}{4} * (-2) + \frac{1}{2} * (-1) \right) = \frac{7}{4} = 1.75 \text{ біт/сим}$$

Ентропія ДВВ Y :

$$H(Y) = -$$

$$\left(\frac{3}{8} \log_2 \frac{3}{8} + \frac{5}{8} \log_2 \frac{5}{8} \right) = -\frac{1}{8} \left(3 \log_2 \frac{3}{8} + 5 \log_2 \frac{5}{8} \right) \approx -\frac{1}{8} (3 * (-1.415) + 5 * (-0.678)) \approx 0.954 \quad ($$

біт/сим)

Побудуємо допоміжну таблицю значень ДВВ $Z = X+Y$ та їх ймовірностей. Оскільки X та Y – незалежні ДВВ, то сумісна ймовірність випадання пар значень (x_i, y_j) :

$$p_{ij} = P(X=i, Y=j) = p(x_i) p(y_j) = p_i q_j, \quad i=0,1,3,4, \quad j=-2,2$$

Сумісна ймовірність випадання пар значень (x_i, y_j)

| Y | X | | | |
|----|------|------|------|------|
| | 0 | 1 | 3 | 4 |
| -2 | -2 | -1 | 1 | 2 |
| | 3/64 | 3/64 | 3/32 | 3/16 |
| 2 | 2 | 3 | 5 | 6 |
| | 5/64 | 5/64 | 5/32 | 5/16 |

Знайдемо ймовірності системи ДВВ $(Z=j, Y=i)$:

$$P(Z=-2, Y=-2) = \frac{3}{64}, \quad P(Z=-1, Y=-2) = \frac{3}{64}, \quad P(Z=1, Y=-2) = \frac{3}{32}, \quad P(Z=2, Y=-2) = \frac{3}{16},$$

$$P(Z=2, Y=2) = \frac{5}{64}, \quad P(Z=3, Y=2) = \frac{5}{64}, \quad P(Z=5, Y=2) = \frac{5}{32}, \quad P(Z=6, Y=2) = \frac{5}{16}.$$

Побудуємо таблицю розподілу ймовірностей системи ДВВ (Y, Z)

Розподіл ймовірностей системи ДВВ (Y, z)

| Y | Z | | | | | | |
|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | -2 | -1 | 1 | 2 | 3 | 5 | 6 |
| -2 | $\frac{3}{64}$ | $\frac{3}{64}$ | $\frac{3}{32}$ | $\frac{3}{16}$ | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | $\frac{5}{64}$ | $\frac{5}{64}$ | $\frac{5}{32}$ | $\frac{5}{16}$ |

Тоді взаємна ентропія ДВВ Z та Y :

$$H(Z, Y) = - \sum_{i,j} p_{ij} \log_2 p_{ij} =$$

$$-(2 \frac{3}{64} \log_2 \frac{3}{64} + \frac{3}{32} \log_2 \frac{3}{32} + \frac{3}{16} \log_2 \frac{3}{16} + 2 \frac{5}{64} \log_2 \frac{5}{64} + \frac{5}{32} \log_2 \frac{5}{32} + \frac{5}{16} \log_2 \frac{5}{16}) \approx$$

$$2.576$$

(біт/сим).

Скориставшись таблицею, побудуємо розподіл ймовірностей ДВВ Z:

Розподіл ймовірностей ДВВ Z:

| Z | -2 | -1 | 1 | 2 | 3 | 5 | 6 |
|-------|------|------|------|-------|------|------|------|
| p_i | 3/64 | 3/64 | 3/32 | 17/64 | 5/64 | 5/32 | 5/16 |

Звідси знаходимо ентропію ДВВ Z:

$$H(Z) =$$

$$-(2 \frac{3}{64} \log_2 \frac{3}{64} + \frac{3}{32} \log_2 \frac{3}{32} + \frac{17}{64} \log_2 \frac{17}{64} + \frac{5}{64} \log_2 \frac{5}{64} + \frac{5}{32} \log_2 \frac{5}{32} + \frac{5}{16} \log_2 \frac{5}{16}) \approx$$

$$2.344$$

біт/сим).

Кількість інформації, що містить ДВВ Z стосовно ДВВ X, знаходимо, скориставшись властивістю кількості інформації і ентропії:

$$I(Z, Y) = H(Z) + H(Y) - H(Z, Y) \approx 2.344 + 0.954 - 2.576 = 0.722 \text{ (біт/сим)}.$$

4. **Відповідь:** $H(X) = 1.75$ (біт/сим), $H(Y) \approx 0.954$ (біт/сим),

$$H(Z) \approx 2.344 \text{ (біт/сим)}, I(Z, Y) \approx 0.722 \text{ (біт/сим)}.$$

Задача 3

1. Постановка завдання

1. Для заданого коду і виразу за індивідуальним завданням скласти таблицю розподілу ймовірностей букв(блоків) алфавіту джерела.
2. Обчислити ентропію джерела.
3. Обчислити надлишковість джерела
4. Побудувати код джерела алфавіту.
5. Обчислити середню довжину коду
6. Обчислити надлишковість отриманого коду..
7. Закодувати заданий вираз отриманим кодом
8. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах і закодованого виразу.

| Варіант | Завдання | Вираз | Код |
|---------|----------|------------------------------------|----------------------------------|
| 12 | 38 | Життя біжить — як музика дзвенить. | Код Шеннона-Фано другого порядку |

9. Оформити звіт з виконання етапів 1-8.

2. Основні поняття

Ентропія джерела – середня кількість інформації, що припадає на одне елементарне повідомлення.

$$H(X) = - \sum_{i=1}^k p_i \log_2 p_i , \quad (1)$$

де $i = 1 \dots k, k$ – об'єм алфавіту джерела(множина символів)

Надлишковість джерела – свідчить про інформаційний резерв повідомлень, елементи яких нерівномірні. Вона показує, яка частка максимально можливої при заданому об'ємі алфавіту невизначеності (ентропії) не використовується джерелом.

$$\rho_{\text{дж}} = 1 - \frac{H(X)}{H(X)_{MAX}} \quad (2)$$

де $H(X)$ – ентропія джерела (1), $H(X)_{MAX} = \log_2 k$ – максимальна ентропія даного джерела.

Середня довжина коду визначається за формулою:

$$\bar{L} = \sum_{i=1}^n l_i p_i \quad (3)$$

Надлишковість коду визначається за формулою:

$$\rho_{\text{коду}} = 1 - \frac{H(X)}{\bar{L}} \quad (4)$$

Алгоритм Шеннона-Фано. Значення ДВВ символів розміщаються у порядку спадання ймовірностей. Потім уся сукупність розділяється на дві приблизно рівні за сумою ймовірностей частини. До коду першої частини додається 0, а до коду другої - 1. Кожна з частин за тим самим принципом знову розділяється (якщо це можливо) на дві частини і т.д.

3. Розв'язання задачі

4. Складаємо таблицю розподілу ймовірностей блоків:

ЖИТЬ_БІЖИТЬ_—_ЯК_МУЗИКА_ДЗВЕНИТЬ

| i | a_i | $p(a_i)$ | i | a_i | $p(a_i)$ |
|-----|-------|----------------|-----|-------|----------------|
| 1 | “ЖИ” | $\frac{2}{17}$ | 9 | “МУ” | $\frac{1}{17}$ |
| 2 | “ТЬ” | $\frac{1}{17}$ | 10 | “ЗИ” | $\frac{1}{17}$ |
| 3 | “ТТ” | $\frac{1}{17}$ | 11 | “КА” | $\frac{1}{17}$ |
| 4 | “Я_” | $\frac{1}{17}$ | 12 | “_Д” | $\frac{1}{17}$ |
| 5 | “БГ” | $\frac{1}{17}$ | 13 | “ЗВ” | $\frac{1}{17}$ |
| 6 | “_—” | $\frac{1}{17}$ | 14 | “ЕН” | $\frac{1}{17}$ |
| 7 | “_Я” | $\frac{1}{17}$ | 15 | “ИТ” | $\frac{1}{17}$ |
| 8 | “К_” | $\frac{1}{17}$ | 16 | “Ь” | $\frac{1}{17}$ |

$$p(a_i) = \frac{n}{N}, \text{де}$$

N – кількість блоків всього,

n – кількість повторів блоку a_i

5. Обчислюємо ентропію джерела за формулою (1).

$$H(X) = - \sum_{i=1}^{16} p_i \log_2 p_i = - \left(\frac{2}{17} \log_2 \frac{2}{17} + \frac{15}{17} \log_2 \frac{1}{17} \right) = 0.1176 * 3.088 + \\ + 0.8824 * 4.088 = 0.3631 + 3.6073 = 3.9704 \text{ біт/символ}$$

6. Обчислюємо надлишковість джерела за формулою (2).

$$\rho_{\text{дк}} = 1 - \frac{H(X)}{H(X)_{\text{MAX}}} = 1 - \frac{3.9704}{\log_2 16} = 1 - \frac{3.9704}{4} = 1 - 0.9926 = 0.0074$$

7. Будуємо код джерела за алгоритмом Шенона-Фано, який описаний в п.2 «Основні поняття»:

| i | a_i | $p(a_i)$ | Код | | | | | l_i | $l_i p_i$ |
|-----|-------|----------------|-----|---|---|---|--|-------|----------------|
| 1 | “ЖИ” | $\frac{2}{17}$ | 0 | 0 | 0 | | | 3 | $\frac{6}{17}$ |
| 2 | “ТЬ” | $\frac{1}{17}$ | 0 | 0 | 1 | 0 | | 4 | $\frac{4}{17}$ |
| 3 | “ТТ” | $\frac{1}{17}$ | 0 | 0 | 1 | 1 | | 4 | $\frac{4}{17}$ |
| 4 | “Я_” | $\frac{1}{17}$ | 0 | 1 | 0 | 0 | | 4 | $\frac{4}{17}$ |
| 5 | “БГ” | $\frac{1}{17}$ | 0 | 1 | 0 | 1 | | 4 | $\frac{4}{17}$ |
| 6 | “_” | $\frac{1}{17}$ | 0 | 1 | 1 | 0 | | 4 | $\frac{4}{17}$ |
| 7 | “_Я” | $\frac{1}{17}$ | 0 | 1 | 1 | 1 | | 4 | $\frac{4}{17}$ |
| 8 | “К_” | $\frac{1}{17}$ | 1 | 0 | 0 | 0 | | 4 | $\frac{4}{17}$ |
| 9 | “МУ” | $\frac{1}{17}$ | 1 | 0 | 0 | 1 | | 4 | $\frac{4}{17}$ |
| 10 | “ЗИ” | $\frac{1}{17}$ | 1 | 0 | 1 | 0 | | 4 | $\frac{4}{17}$ |
| 11 | “КА” | $\frac{1}{17}$ | 1 | 0 | 1 | 1 | | 4 | $\frac{4}{17}$ |
| 12 | “_Д” | $\frac{1}{17}$ | 1 | 1 | 0 | 0 | | 4 | $\frac{4}{17}$ |
| 13 | “ЗВ” | $\frac{1}{17}$ | 1 | 1 | 0 | 1 | | 4 | $\frac{4}{17}$ |

| | | | | | | | | | |
|--------|------|----------------|---|---|---|---|---|---|-----------------|
| 14 | “ЕН” | $\frac{1}{17}$ | 1 | 1 | 1 | 0 | | 4 | $\frac{4}{17}$ |
| 15 | “ИТ” | $\frac{1}{17}$ | 1 | 1 | 1 | 1 | 0 | 5 | $\frac{5}{17}$ |
| 16 | “Ь” | $\frac{1}{17}$ | 1 | 1 | 1 | 1 | 1 | 5 | $\frac{5}{17}$ |
| \sum | | 1 | | | | | | | $\frac{68}{17}$ |

8. Обчислюємо середню довжину коду:

$$\bar{L} = \sum_{i=1}^n l_i p_i = \frac{6}{17} + 13 * \frac{4}{17} + 2 * \frac{5}{17} = \frac{68}{17} = 4 \text{ біти/символ}$$

9. Обчислюємо надлишковість отриманого коду:

$$\rho_{\text{коду}} = 1 - \frac{3.9704}{\bar{L}} = \frac{3.9704}{4} = 0.0074$$

10. Кодуємо даний вираз знайденим кодом:

| ЖИ | ТТ | Я_ | БІ | ЖИ | ТЬ | _Я | К_ |
|------|------|------|------|------|------|-------|-------|
| 000 | 0011 | 0100 | 0101 | 000 | 0010 | 0111 | 1000 |
| МУ | ЗИ | КА | Д | ЗВ | ЕН | ИТ | Ь |
| 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 11110 | 11111 |

11. Порівнюємо довжини заданого за індивідуальним завданням виразу в ASCII-кодах і закодованого виразу:

Довжина даного виразу в ASCII коді: $33 * 8 = 264$ біт

Довжина даного виразу в коді Шенона-Фано другого порядку: 64 біт

$$\frac{264}{64} * 100\% = 412.5\%$$

Отже, код Шенона-Фано в даній ситуації на 412.5 % коротший за ASCII.

Задача 4

Постановка задачі

1. Для заданого коду і виразу за індивідуальним завданням скласти таблицю розподілу ймовірностей букв(блоків) алфавіту джерела.
2. Обчислити ентропію джерела.
3. Обчислити надлишковість джерела
4. Побудувати код джерела алфавіту.
5. Обчислити середню довжину коду.
6. Обчислити надлишковість отриманого коду.
7. Закодувати заданий вираз отриманим кодом.
8. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах і закодованого виразу.
9. Оформити звіт з виконання етапів 1-8

| Варіант | Завдання | Вираз | Код |
|---------|----------|--------------------------------|---------------------------------|
| 7 | 43 | Навчай інших — і сам навчишся. | Код Хаффмена першого порядку |

Загальні формули

Для розв'язку задачі використовуємо такі формули:

$$12. \text{ Ентропія джерела: } H(X) = - \sum_{i=1}^k p_i \log_2 p_i ,$$

де k - об'єм алфавіту джерела для незалежних повідомлень.

$$13. \text{ Надлишковість джерела: } \rho_{\text{дж}} = 1 - \frac{H(x)}{H_{\max}(x)} .$$

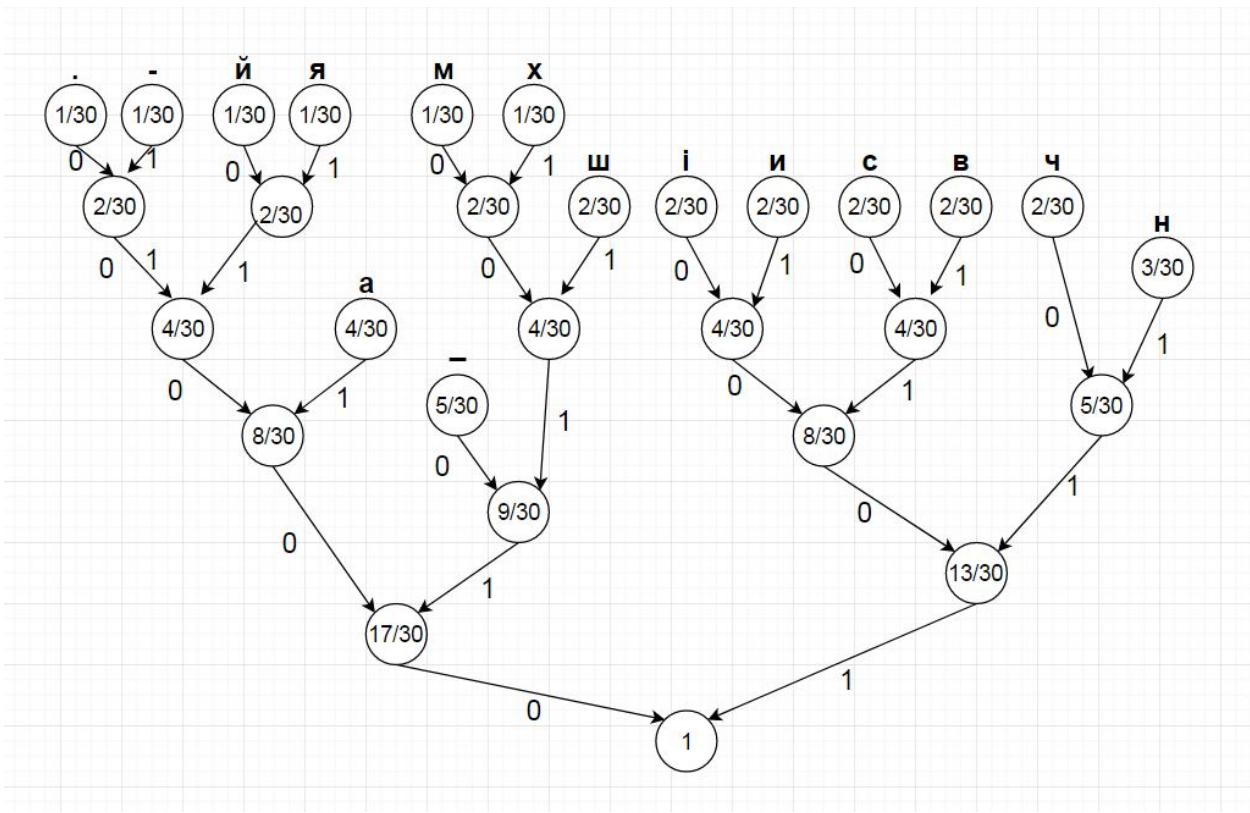
$$14. \text{ Середня довжина коду: } \bar{l} = \sum_{i=1}^k l_i p_i .$$

$$15. \text{ Надлишковість коду: } \rho_k = 1 - \frac{H(x)}{\bar{l}} .$$

Розв'язок задачі

Побудуємо кодове дерево:

Де $_$ - означає пробіл.



Побудуємо таблицю коду Хаффмена третього порядку:

| Знак, a_i | Ймовірність , p_i | Код | l_i – довжина коду | $p_i l_i$ |
|----------------|------------------------|-------|-------------------------|----------------|
| . | $\frac{1}{30}$ | 00000 | 5 | $\frac{5}{30}$ |
| - | $\frac{1}{30}$ | 00001 | 5 | $\frac{5}{30}$ |
| й | $\frac{1}{30}$ | 00010 | 5 | $\frac{5}{30}$ |
| м | $\frac{1}{30}$ | 01100 | 5 | $\frac{5}{30}$ |
| х | $\frac{1}{30}$ | 01101 | 5 | $\frac{5}{28}$ |
| я | $\frac{1}{30}$ | 00011 | 5 | $\frac{5}{28}$ |
| с | $\frac{2}{30}$ | 1010 | 4 | $\frac{8}{30}$ |
| и | $\frac{2}{30}$ | 1001 | 4 | $\frac{8}{30}$ |

| | | | | |
|-----|----------------|------|---|-----------------|
| ІІІ | $\frac{2}{30}$ | 0111 | 4 | $\frac{8}{30}$ |
| I | $\frac{2}{30}$ | 1000 | 4 | $\frac{8}{30}$ |
| Ч | $\frac{2}{30}$ | 110 | 3 | $\frac{6}{30}$ |
| В | $\frac{2}{30}$ | 1011 | 4 | $\frac{8}{30}$ |
| Н | $\frac{3}{30}$ | 111 | 3 | $\frac{9}{30}$ |
| A | $\frac{4}{30}$ | 001 | 3 | $\frac{12}{30}$ |
| - | $\frac{5}{30}$ | 010 | 3 | $\frac{15}{30}$ |

$$\text{Ентропія джерела: } \frac{1}{28} + \frac{3*3}{28} \log_2 \frac{1}{28} + \frac{4}{28} \log_2 \frac{1}{28} + \frac{5}{28} \log_2 \frac{1}{28} + \frac{6*2}{28} \log_2 \frac{1}{28} \approx 3,936 \left(\frac{\text{біт}}{\text{симв}} \right)$$

$$H(x) = -\sum p_i \log_2 p_i$$

$$\text{Надлишковість джерела: } \rho_{\text{ож}} = 1 - \frac{H(x)}{\log_2 28} = 0.181.$$

$$\text{Середня довжина коду: } \bar{l} = \sum_{i=1}^{17} p_i l_i \approx 3.967.$$

$$\text{Надлишковість коду: } \rho_{\kappa} = 1 - \frac{3.936}{3.967} \approx 0.00781.$$

Закодуємо наш вираз:

111 001 1011 110 001 00010 010 1000 111 0111 1001 01101 010 00001 010 1000
010 1010 001 01100 010 111 001 1011 110 1001 0111 1010 00011 00000

Довжина даного коду: 104 біт.

Довжина даного повідомлення в коді ASCII: 240 біт.

Задача 5

Завдання № 37

1. Для заданого **коду і виразу** за індивідуальним завданням скласти таблицю розподілу ймовірностей букв(блоків) алфавіту джерела.
2. Обчислити ентропію джерела.
3. Обчислити надлишковість джерела
4. Побудувати код джерела алфавіту.
5. Обчислити середню довжину коду
6. Обчислити надлишковість отриманого коду.
7. Закодувати заданий вираз отриманим кодом
8. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах і закодованого виразу.

| | |
|--|----------------------------------|
| За одного вченого дають десять невчених. | Код Шеннона-Фано другого порядку |
|--|----------------------------------|

Розв'язок задачі

Метод Шеннона-Фано.

1. Розмістимо значення ДВВ символів у порядку спадання ймовірностей.

Потім розділимо вся сукупність на дві приблизно рівні за сумою ймовірностей частини. До коду першої частини додамо 0, а до коду другої - 1. Кожну з частин за тим самим принципом знову розділяємо (якщо це можливо) на дві частини і т.д.

Складемо таблицю розподілу ймовірностей букв(блоків) алфавіту джерела та визначимо код джерела алфавіту

| Буква х _i | Ймовірність | Код | Довжина | p _i l _i |
|----------------------|----------------|-------|---------|-------------------------------|
| о _Г | $\frac{1}{10}$ | 000 | 3 | $\frac{3}{10}$ |
| о __ | $\frac{1}{10}$ | 0010 | 4 | $\frac{4}{10}$ |
| з _а | $\frac{1}{20}$ | 0011 | 4 | $\frac{4}{20}$ |
| — _о | $\frac{1}{20}$ | 01000 | 5 | $\frac{5}{20}$ |
| д _н | $\frac{1}{20}$ | 01001 | 5 | $\frac{5}{20}$ |
| в _ч | $\frac{1}{20}$ | 0101 | 4 | $\frac{4}{20}$ |
| е _н | $\frac{1}{20}$ | 0110 | 4 | $\frac{4}{20}$ |
| да | $\frac{1}{20}$ | 0111 | 4 | $\frac{4}{20}$ |
| ю _т | $\frac{1}{20}$ | 10000 | 5 | $\frac{5}{20}$ |
| ь __ | $\frac{1}{20}$ | 10001 | 5 | $\frac{5}{20}$ |
| де | $\frac{1}{20}$ | 1001 | 4 | $\frac{4}{20}$ |
| ся | $\frac{1}{20}$ | 1010 | 4 | $\frac{4}{20}$ |
| ть | $\frac{1}{20}$ | 1011 | 4 | $\frac{4}{20}$ |
| — _н | $\frac{1}{20}$ | 11000 | 5 | $\frac{5}{20}$ |
| ев | $\frac{1}{20}$ | 11001 | 5 | $\frac{5}{20}$ |
| че | $\frac{1}{20}$ | 1101 | 4 | $\frac{4}{20}$ |
| ни | $\frac{1}{20}$ | 1110 | 4 | $\frac{4}{20}$ |
| х. | $\frac{1}{20}$ | 1111 | 4 | $\frac{4}{20}$ |

2. Обчислимо ентропію джерела за наступною формулою.

$$\text{Ентропія джерела: } H(X) = - \sum_{i=1}^k p_i \log_2 p_i, \quad i=1, \dots, k$$

$$H(X) = - \left(\frac{2*1}{10} \log_2 \frac{1}{10} + \frac{16*1}{20} \log_2 \frac{1}{20} \right) \approx -2*0.1*(-3.321) - 16*0.05*(-4.321) = 0.6642 + 3.4568 = 4.121 \text{ біт/сим}$$

3. Обчислимо надлишковість джерела

$$\rho_x = 1 - \frac{H(X)}{H(X)_{\max}} = 1 - \frac{H(X)}{\log_2 k},$$

де $H(X)$ - ентропія джерела повідомлень;

$H(X)_{\max} = \log_2 k$ - максимальна досяжна ентропія даного джерела

$$\rho_x = 1 - \frac{8.6908}{(\log_2 18)} = 1 - \frac{4.121}{4.169} = 1 - 0.9885 = 0.0115$$

4. Обчислимо середню довжину коду

$$E = \sum_{i=1}^{18} p_i l_i = \frac{3}{10} + \frac{4}{10} + \dots + 10 * \frac{4}{20} + 6 * \frac{5}{20} = 4.2 \text{ (біт/сим);}$$

5. Обчислимо надлишковість отриманого коду

$$\rho_k = 1 - \frac{H(X)}{E}$$

$$\rho_k = 1 - \frac{4.121}{4.2} = 1 - 0.9811 = 0.0189$$

Закодуємо заданий вираз отриманим кодом

0011 01000 01001 000 0010 0101 0110 000 0010 0111 10000 10001 1001
1010 1011 11000 11001 1101 1110 1111

Довжина закодованного повідомлення за допомогою кода Шеннона-Фано складає 85 символи, за допомогою ASCII-коду – $40 * 8 = 320$. Отже, кодування методом Шеннона-Фано значно компактніше.

$$\text{Коефіцієнт стискання } \frac{320}{85} = 3.76$$

Відповідь: Коефіцієнт стискання $\frac{320}{85} = 3.76$

ентропія $H(X) = 4.121$ (біт/сим);

надлишковість джерела $\rho_x = 0.0115$;

середня довжина коду $E = 4.2$ (біт/сим);

надлишковість отриманого коду $\rho_k = 0.0189$.

Задача 6

1. Постановка задачі

1. Для заданого **коду** (Адаптивний алгоритм Хаффмена) і **виразу** (Spes vana) за алгоритмом індивідуального завдання (Адаптивний алгоритм Хаффмена) побудувати таблицю кодування (закодувати) повідомлення виразу.
2. Обчислити ентропію джерела.
3. Виконати декодування закодованого за пунктом 1 повідомлення виразу.
4. Побудувати дерево кодування **виразу** індивідуального завдання кодом Хаффмена.
5. Закодувати заданий **вираз** отриманим кодом Хаффмена
6. Обчислити довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим **кодом** (алгоритмом) і кодом Хаффмена
7. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим кодом (алгоритмом) і кодом Хаффмена. Зробити висновки
8. Оформити звіт з виконання етапів 1-7.

| | | | |
|--------|-----------|--------------|---------------------------------|
| 8 5 | Spes vana | Марна надія. | Адаптивний алгоритм Хаффмена |
|--------|-----------|--------------|---------------------------------|

2. Математичний розв'язок задачі

1. Для заданого коду і виразу за алгоритмом індивідуального завдання побудувати таблицю кодування (закодувати) повідомлення виразу. Для арифметичного алгоритму визначити таблицю розподілу символів.

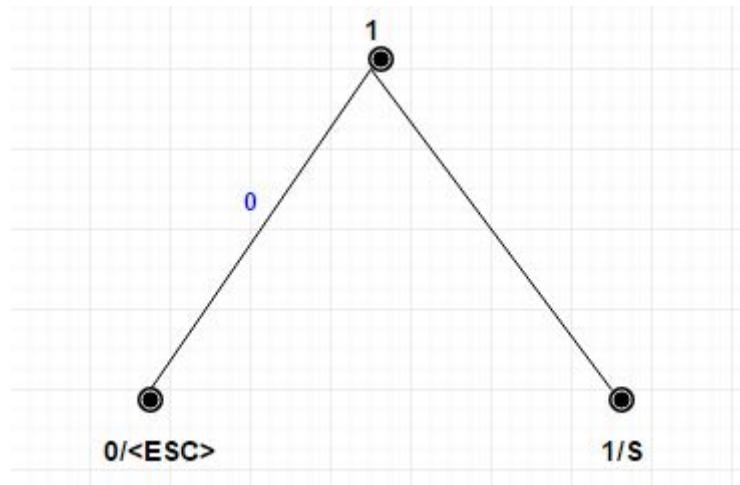
В адаптивному (динамічному) алгоритмі Хаффмана таблиця кодів не передається. У процесі кодування залежно від значення поточного символу, що надходять на вход алгоритму, кодове дерево коригується відповідно до зміни статистики вхідного потоку. При декодуванні відбувається той самий процес. Для однозначності декодування використовується упорядкована структура кодового дерева. Упорядкованим деревом Хаффмана називається бінарне дерево, вузли якого можуть бути перелічені у порядку неубування їх ваги зліва-направо на кожному рівні і знизу-вверх за рівнями. При зміні ваги існуючого вузла в дереві достатньо помінити місцями два вузли: вузол, що порушив упорядкованість, і останній з наступних за ним вузлів меншої ваги. Після обміну вузлів місцями необхідно перерахувати вагу всіх вузлів-предків. На початку роботи алгоритму дерево містить тільки один спеціальний символ <ESC>, що завжди має частоту 0. Він необхідний для занесення в кодове дерево нового символу, що передається безпосередньо після <ESC>. При появи

нового символу праворуч від вузла <ESC> додається лист і потім, якщо необхідно, дерево упорядковується. Ліві гілки кодового дерева позначаються 0, а праві – 1. Спочатку зробимо опис потоку вхідних символів

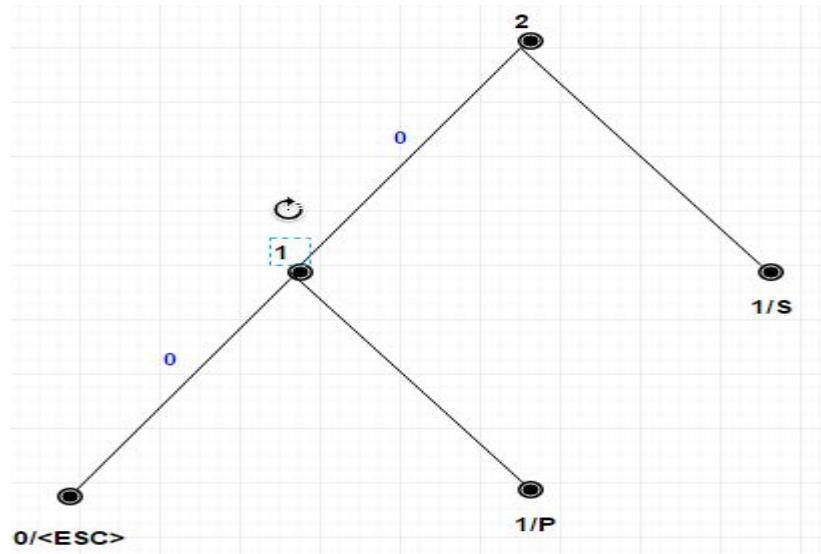
| Вхідні дані | Код | Довжина коду | Номер дерева |
|-------------|---------|--------------|--------------|
| S | 'S' | 8 | 1 |
| P | 0'P' | 9 | 2 |
| E | 00'E' | 10 | 3 |
| S | 0 | 1 | 4 |
| | 100' ,' | 11 | 5 |
| V | 1100'V' | 12 | 6 |
| A | 1000'A' | 12 | 7 |
| N | 0100'N' | 12 | 8 |
| A | 001 | 3 | 9 |

За вище заданим принципом побудуємо дерева для кожного з символів відповідно.

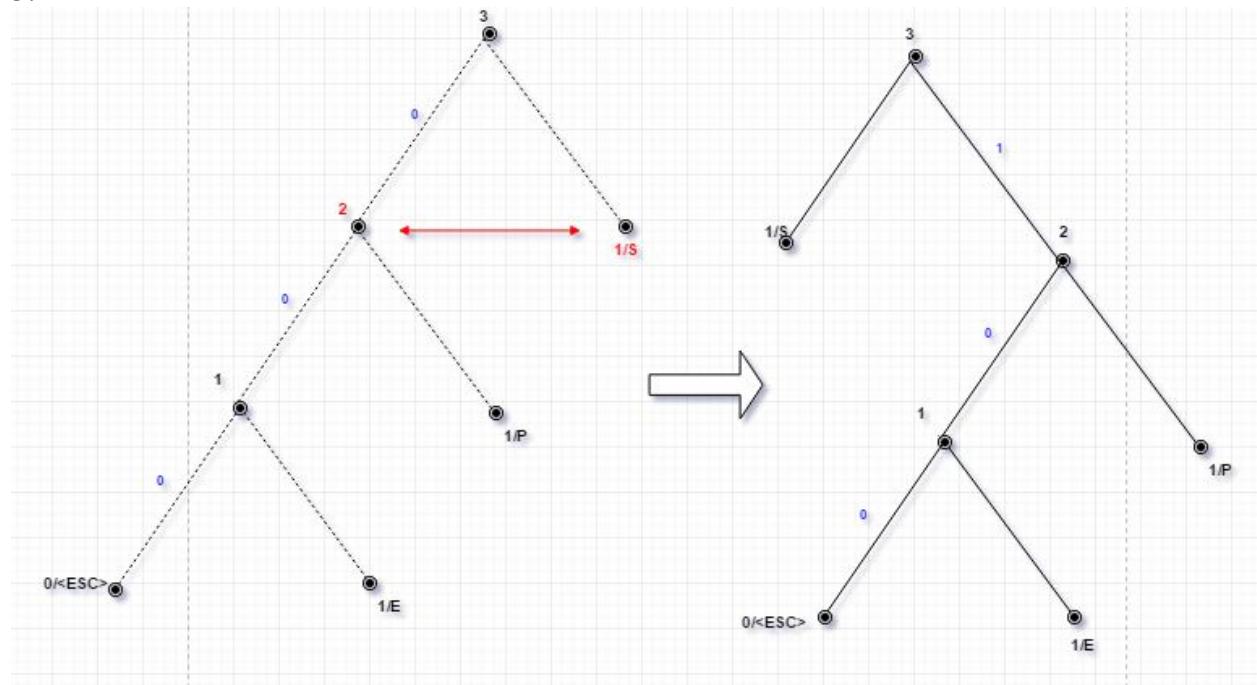
1.



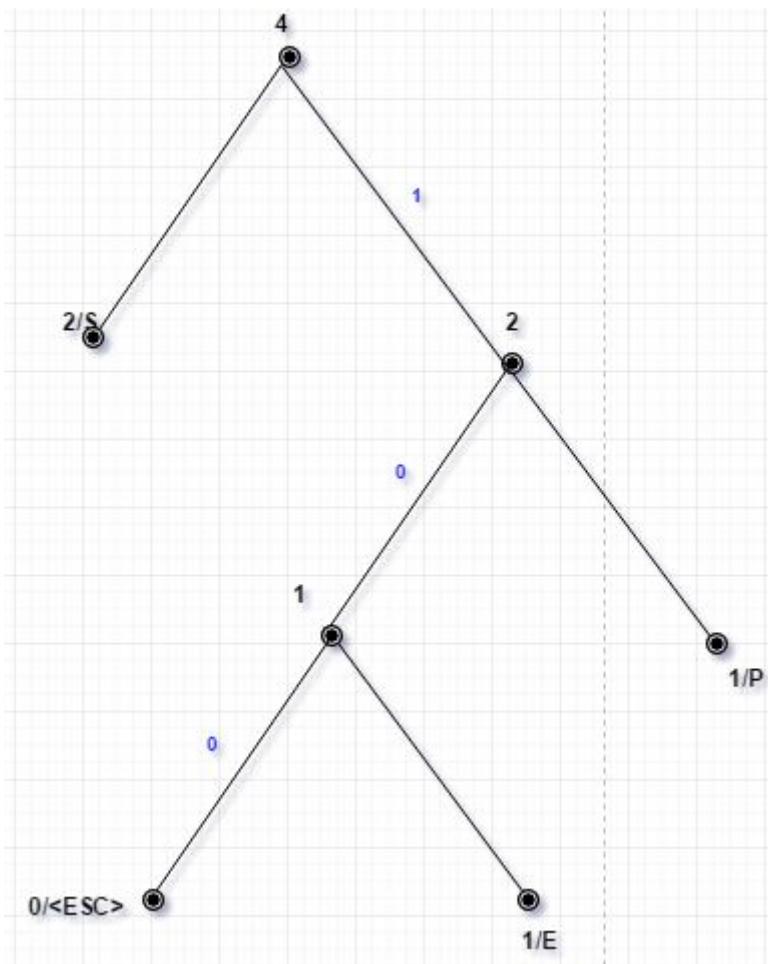
2.



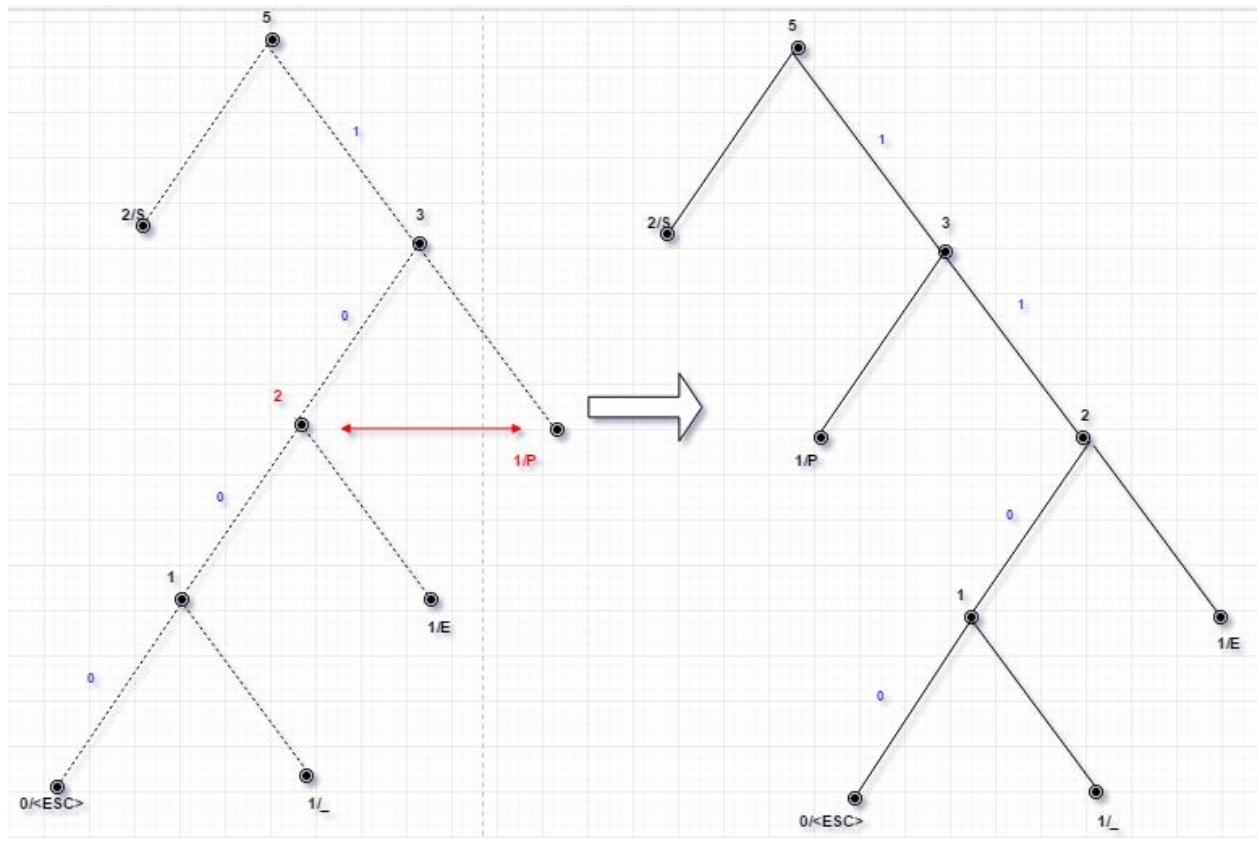
3.



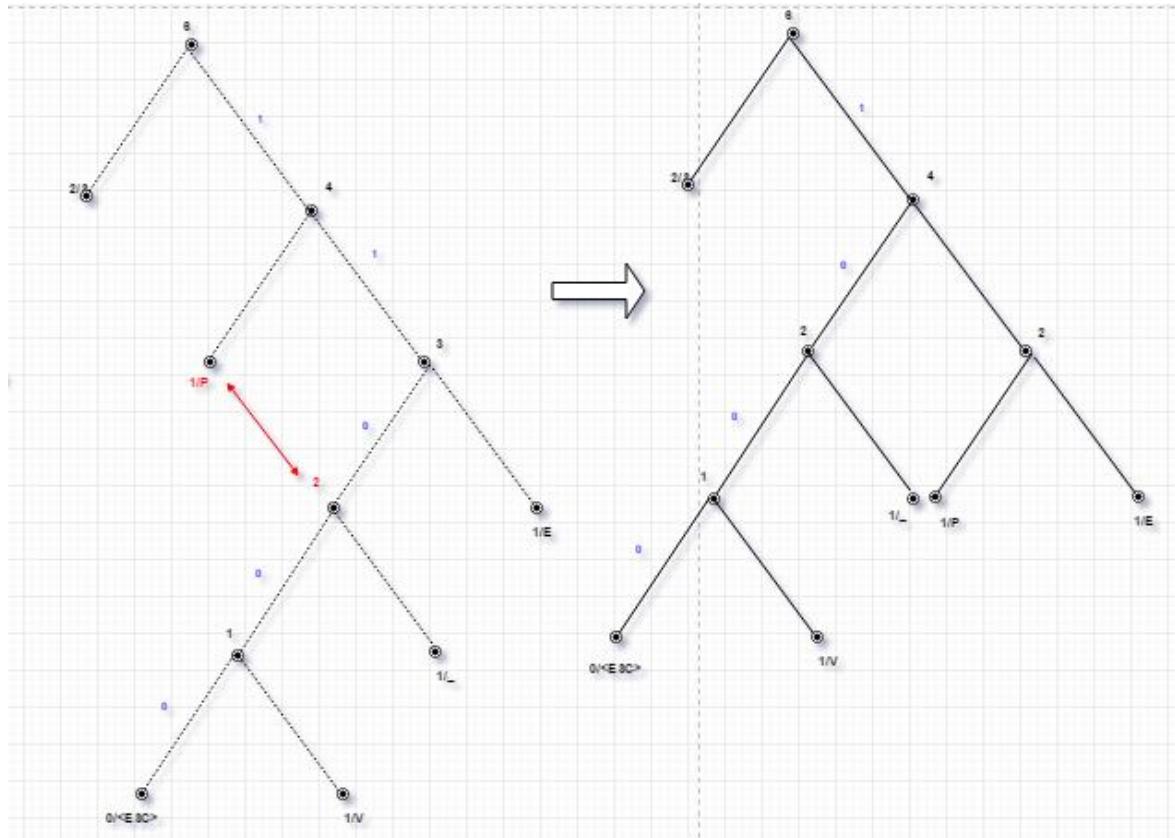
4.



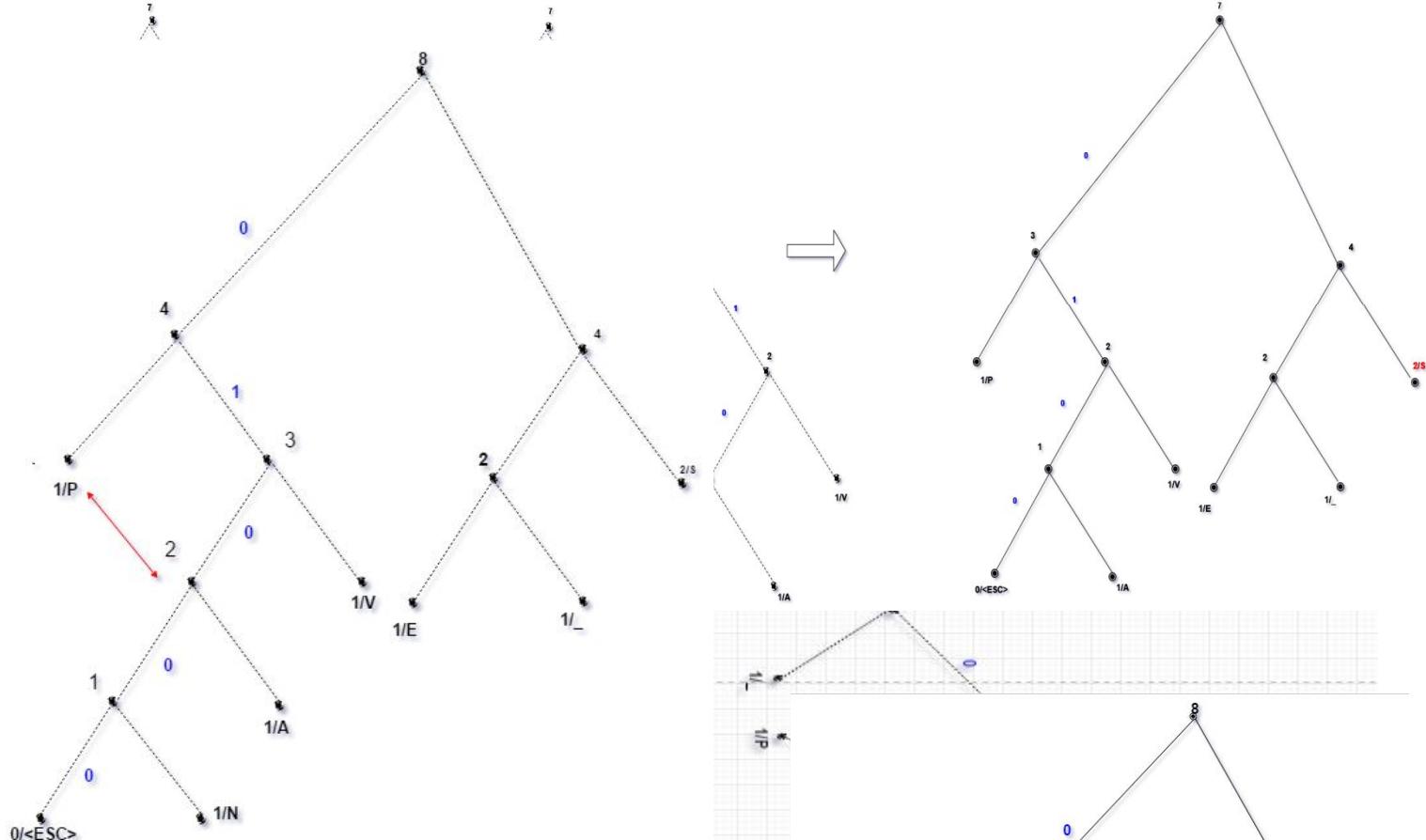
5.



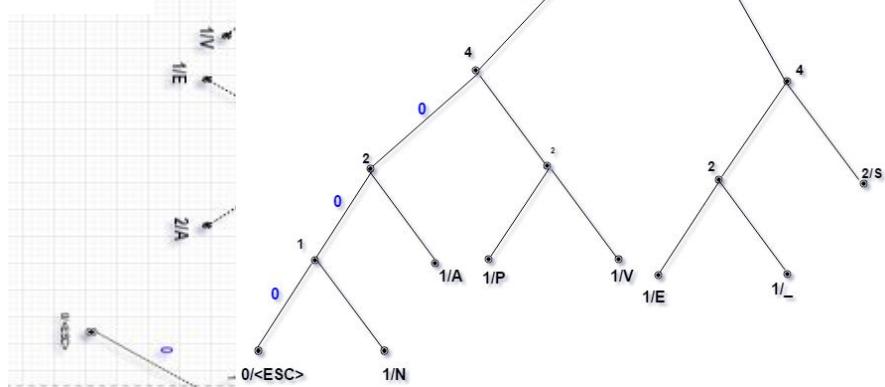
6.



7.

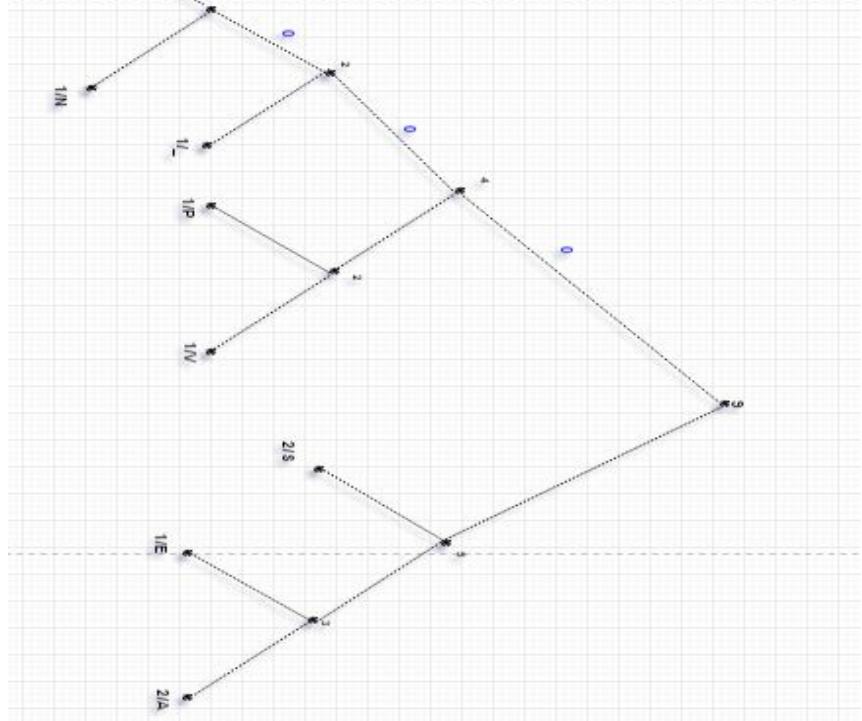


8.



9.

Повідомлення
кодується
безпосередньо



послідовністю кодів для кожного символу.

Закодоване повідомлення : ‘S’0’P’00’E’0100’_’1100’V’1000’A’0100”n”001

2. Обчислити ентропію джерела.

Кількість інформації джерела дискретних повідомлень x_i , оцінюють середньою кількістю інформації, що припадає на одне елементарне повідомлення, як ентропією джерела за формулою:

$$H(X) = - \sum_{i=1}^k p_i \log_2 p_i \quad i=1..k,$$

де k - об'єм алфавіту джерела для незалежних повідомлень.

Для знаходження ентропії необхідно порахувати ймовірності для кожного символу повідомлення, для зручності зробимо таблицю:

Ймовірність рахуємо, як кількість певного символа до загальної кількості символів у повідомленні.

| Символ | S | P | E | <u> </u> | V | A | N |
|-------------|-----|-----|-----|----------|-----|-----|-----|
| Ймовірність | 2/9 | 1/9 | 1/9 | 1/9 | 1/9 | 2/9 | 1/9 |

Знайдемо ентропію ДВВ X, за формулою наведеною вище:

$$H(X) = -(2*2/9 * \log_2 \frac{2}{9} + 5*1/9 * \log_2 1/9) = -(\frac{2}{9} * \log_2 \frac{2}{9} + \frac{5}{9} * \log_2 \frac{1}{9}) = -(\frac{2}{9} * \log_2 9 - \frac{4}{9} + \frac{5}{9} * \log_2 9) = -(\frac{2}{9} * 4 + \frac{5}{9} * 4) = -(\frac{18}{9}) = -2 = 2,73 \text{ біт/символ}$$

2. Виконати декодування закодованого за пунктом 1 повідомлення виразу.

Нам необхідно декодувати повідомлення

‘S’0’P’00’E’0100’_’1100’V’1000’A’0100”n”001, закодоване за адаптивним алгоритмом Хаффмана.

Декодування відбувається безпосередньо за заданим кодом за допомогою побудови дерев.

Відбувається за правилами:

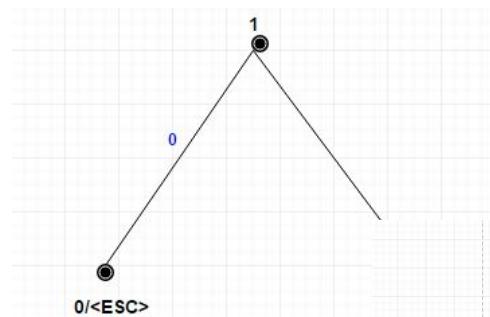
1. Елементи вхідного повідомлення читаються побітно
2. Кожний раз при зчитуванні 0 або 1 відбувається переміщення від кореня вниз по відповідній гілці бінарного дерева Хаффмана, до того часу, поки не буде досягнутий будь який, так званий листок дерева.
3. Якщо досягнутий листок, відповідний символ записується у вихідне повідомлення. Вага листка збільшується на один, вага вузлів-предків корегується, дерево за необхідністю впорядковується.

4. Якщо ж досягнутий escape-символ, то у дерево записується новий символ, вага вузлів-предків корегуються, потім за необхідністю відбувається його впорядкування.

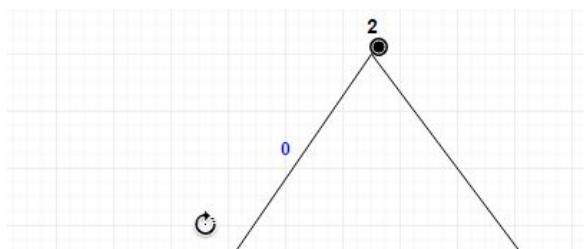
Розглянемо сам процес декодування повідомлення

'S'0'P'00'E'0100' _ '1100'V'1000'A'0100"n"001. На початку декодування дерево Хаффмана містить тільки esc символ з частотою 0.3 розкодуванням кожного нового символу дерево перебудовується, а на виході отримуємо послідовність.

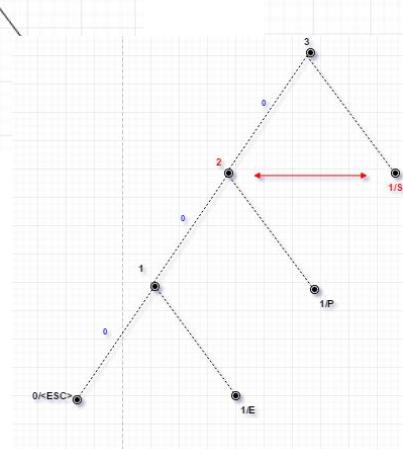
1.



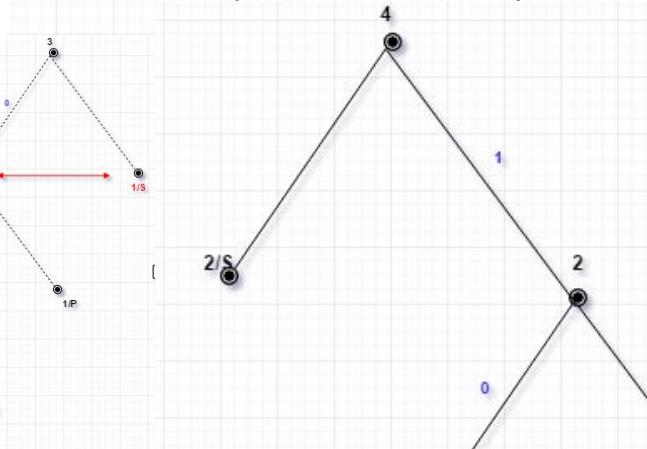
2.



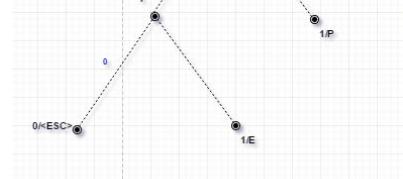
3.



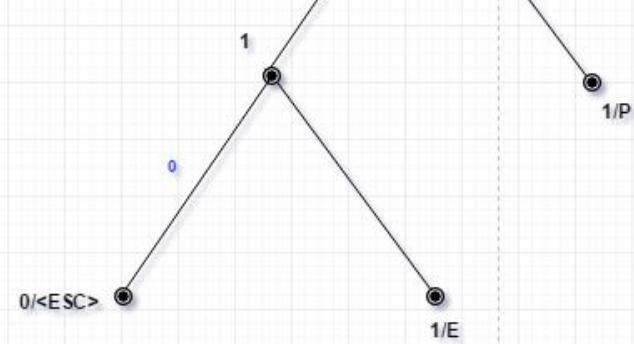
2.

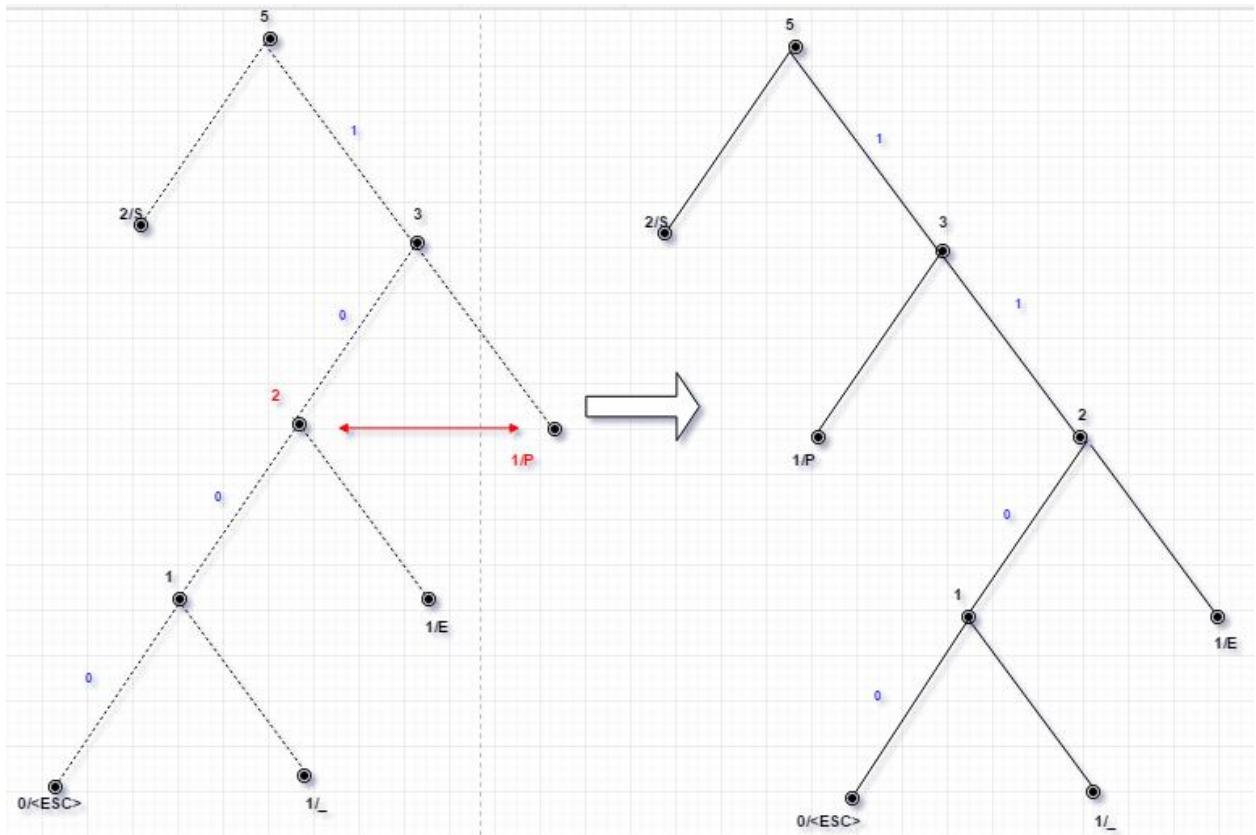


4.

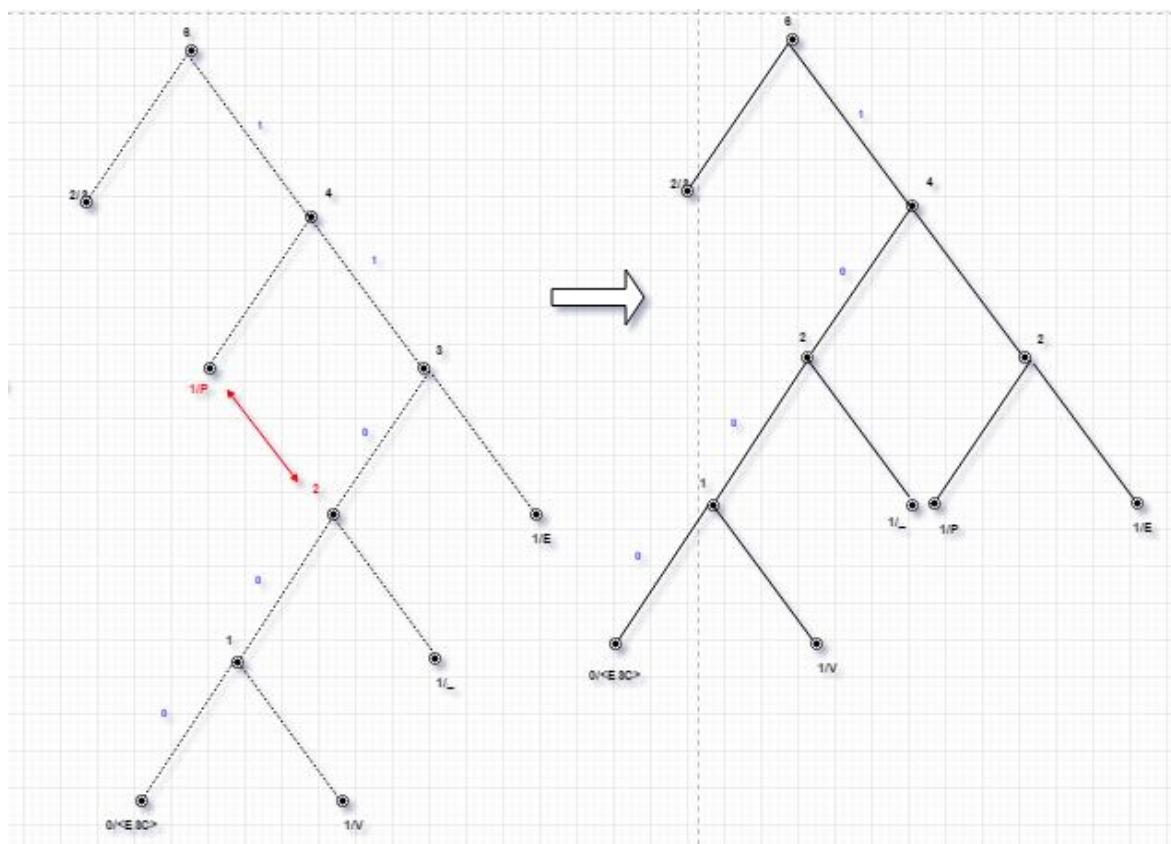


5.

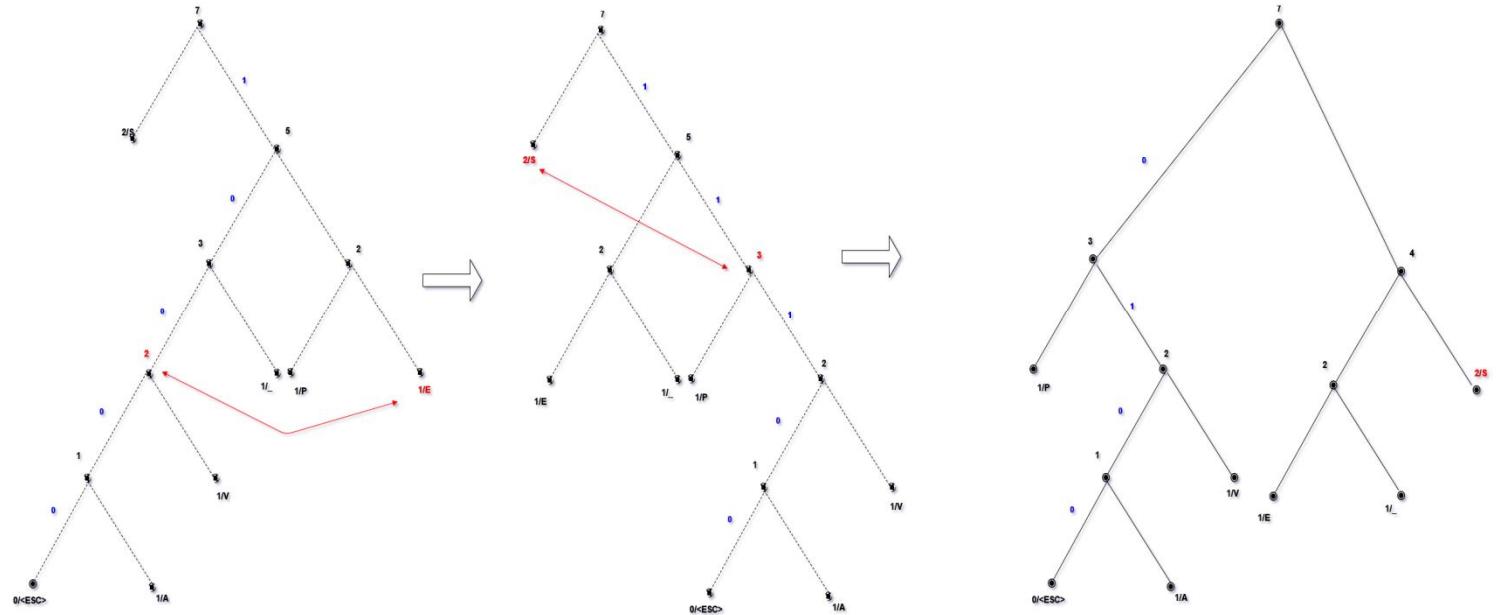




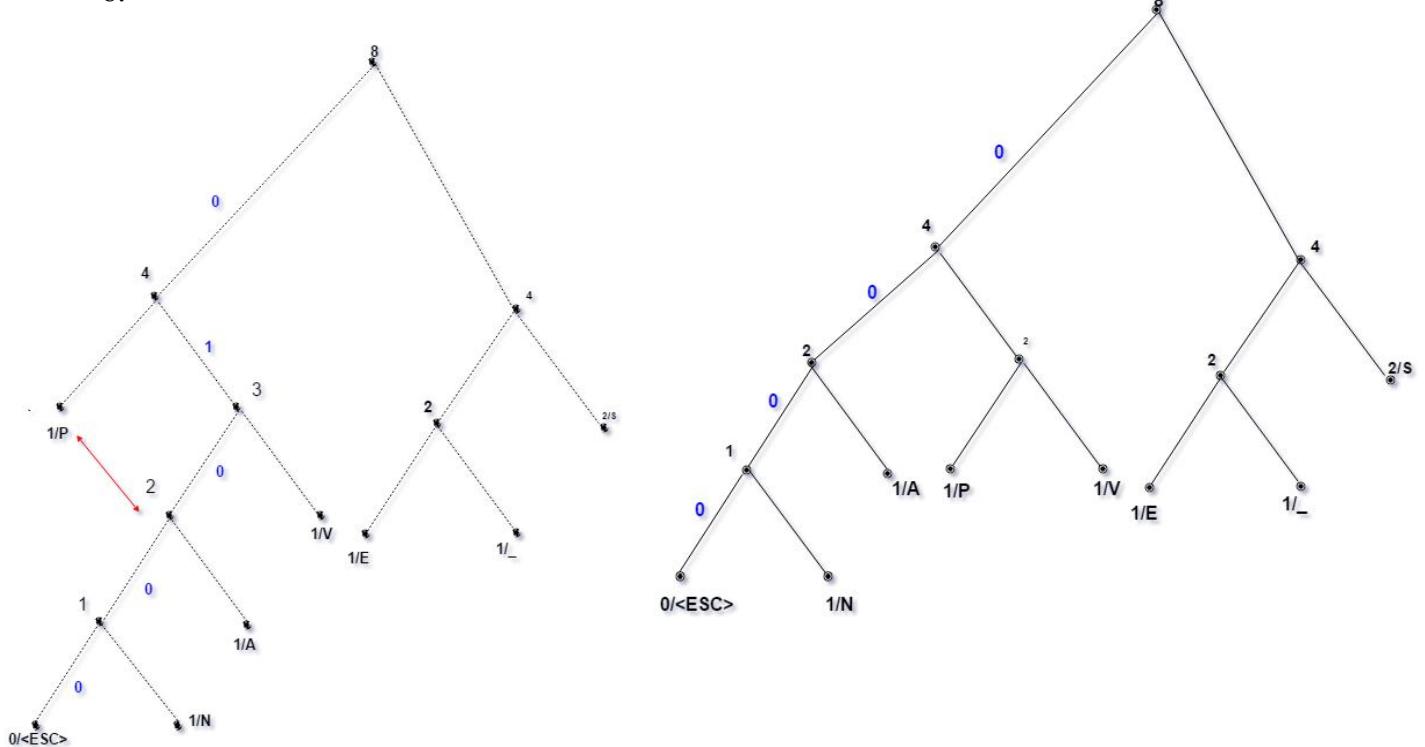
6.



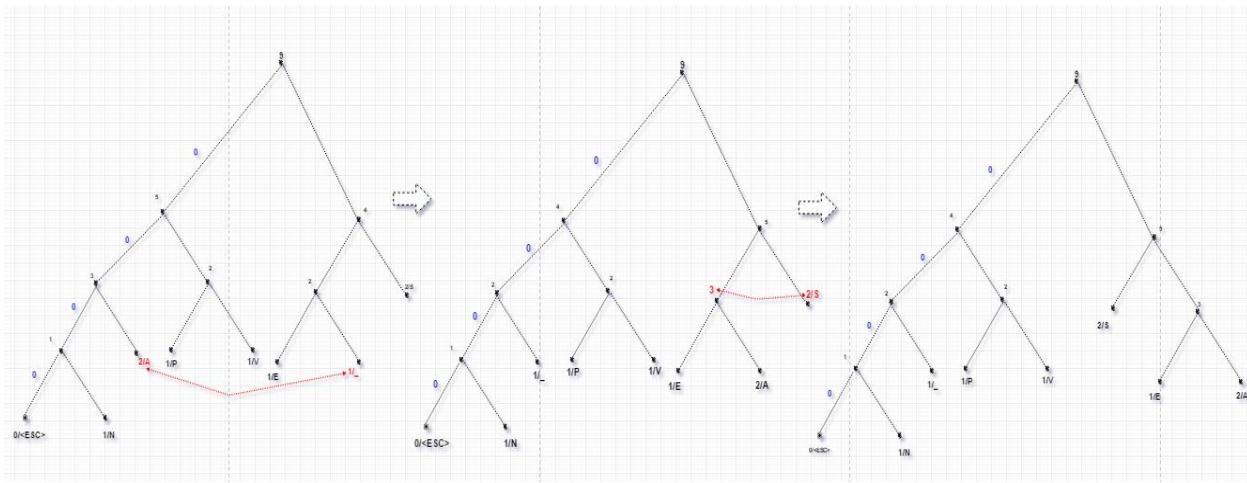
7.



8.



9.



Приклад таблиці декодування

| Вхідні дані | Код | Довжина коду | Номер дерева |
|-------------|-----|--------------|--------------|
| 'S' | S | 8 | 1 |
| 0'P' | P | 9 | 2 |
| 00'E' | E | 10 | 3 |
| 0 | S | 1 | 4 |
| 100' ' | — | 11 | 5 |
| 1100'V' | V | 12 | 6 |
| 1000'A' | A | 12 | 7 |
| 0100'N' | N | 12 | 8 |
| 001 | A | 3 | 9 |

Задача 7

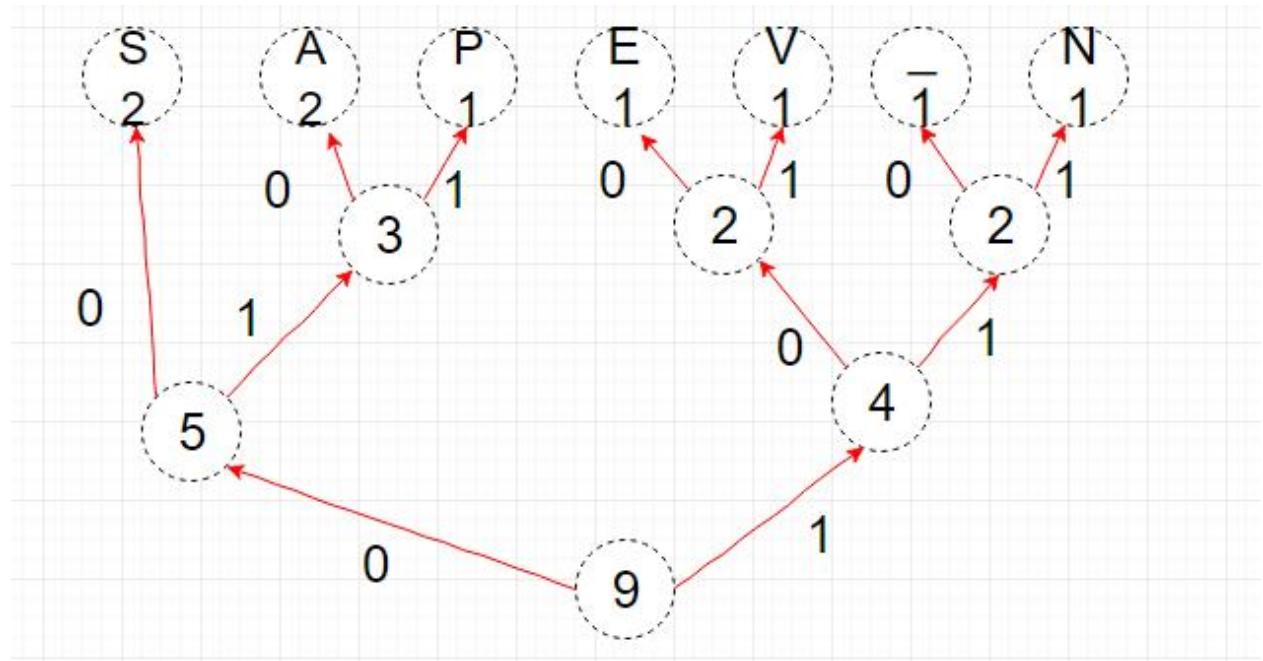
4. Побудувати дерево кодування виразу індивідуального завдання кодом Хаффмена.

Складемо таблицю розподілу ймовірностей букв алфавіту джерела.

Spes_vana

| S | P | E | | V | A | N |
|-----|-----|-----|-----|-----|-----|-----|
| 2/9 | 1/9 | 1/9 | 1/9 | 1/9 | 2/9 | 1/9 |

Алгоритм створення кода полягає у побудові бінарного дерева. Ймовірності ДВВ розміщаються у порядку спадання і приписують листам кодового дерева. Величину, що приписують вузлу дерева, називають його вагою. Два листи або вузли з найменшими значеннями ваги утворюють батьківський вузол, вага якого дорівнює сумарній вазі вузлів, що його складають. Надалі цей вузол враховується нарівні з вершинами, що залишилися, а листя або вузли, що його утворили, більше не розглядаються. Після побудови кореня кожна визначена гілка, що виходить з батьківського вузла, позначається 0 (зазвичай це ліва гілка) або 1 (права гілка). Коди значень ДВВ – це послідовності 0 і 1, що утворюються, на шляху від кореня кодового дерева до листа із заданою імовірністю ДВВ.



| Буква, x_i | Ймовірність, p_i | Код, $Code(X_i)$ | Довжина коду, l_i | lipi |
|--------------|--------------------|------------------|---------------------|------|
| | | | | |

| | | | | |
|-----------|----------------|-----|---|---|
| S | 2/9 | 00 | 2 | 4/9 |
| A | 2/9 | 010 | 3 | 6/9 |
| N | 1/9 | 111 | 3 | 3/9 |
| P | 1/9 | 011 | 3 | 3/9 |
| E | 1/9 | 100 | 3 | 3/9 |
| | 1/9 | 110 | 3 | 3/9 |
| \bar{V} | 1/9 | 101 | 3 | 3/9 |
| | $\sum p_i = 1$ | | | $\sum p_i l_i = \frac{25}{9} = 2,7$ біт/символ |

5. Закодувати заданий вираз отриманим кодом Хаффмена

Код однозначно визначається кодуванням символів

S P E S V A N A
00 011 100 00 110 101 010 111 010

6. Обчислити довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим кодом (алгоритмом) і кодом Хаффмена

Довжина вхідного повідомлення 9 символів.

Довжина закодованого повідомлення адаптивним алгоритмом Хаффмена:

Для цього необхідно підрахувати загальну кількість бітів

З таблиці отримуємо

| Вхідні дані | Код | Довжина коду | Номер дерева |
|-------------|---------|--------------|--------------|
| S | 'S' | 8 | 1 |
| P | 0'P' | 9 | 2 |
| E | 00'E' | 10 | 3 |
| S | 0 | 1 | 4 |
| | 100' | 11 | 5 |
| \bar{V} | 1100'V' | 12 | 6 |
| A | 1000'A' | 12 | 7 |
| N | 0100'N' | 12 | 8 |
| A | 001 | 3 | 9 |

1 буква=8 біт

$L = 8+9+10+1+11+12+12+12+3 = 78$ біт

Довжина повідомлення закодованого кодом Хаффмена

$L=25$ біт

Довжина ASCII

$L=9*8=72$ біт

7. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим кодом (алгоритмом) і . кодом Хаффмена. Зробити висновки

Необхідно було закодувати повідомлення Spes_vana. Із заданих алгоритмів саме для кодування цього виразу найкращим виявився код Хаффмена, так як довжина повідомлення складає 23 біти. Другим за ефективністю виявився код ASCII, а третім адаптивний алгоритм Хаффмана. Мое повідомлення складаєсього на всього 9 символів.

Найбільш ефективним у даному випадку виявився код Хаффмена так як його довжина займає у $78/25 \approx 3$ рази менше чим довжина коду закодованим алгоритмом. Більш ефективним виявився і ASCII код.

Однак у загальному випадку, при наявності більшої кількості символів, найбільш ефективним є адаптивний алгоритм Хаффмана(у нашому випадку ми виграємо лише на 2 символах, а всі інші символи є унікальними), після нього йде код Хаффмана і менш ефективним ASCII код.

Варіант 9 завдання 82

Постановка задачі

Вираз: Laboremus!

Код: Арифметичний алгоритм

9. Для заданого коду і виразу за алгоритмом індивідуального завдання (Арифметичний алгоритм) закодувати повідомлення виразу. Для арифметичного алгоритму визначити таблицю розподілу символів.

Загальні формули

$Low_i = low_{i-1} + \Delta low(s)$ – обчислення лівої границі в арифметичному алгоритмі.

$High_i = low_{i-1} + \Delta high(s)$ – обчислення правої границі в арифметичному алгоритмі.

Основні поняття

За розподілом ймовірностей ДВВ складається таблиця з пересічних в граничних точках відрізків для кожного із значень ДВВ. Об'єднання цих відрізків утворює інтервал $[0;1]$, а їх довжини пропорційні ймовірностям значень ДВВ. Алгоритм кодування полягає в побудові інтервалу, що однозначно визначає конкретну послідовність значень ДВВ. Інтервали повідомлення будуються так: якщо є відрізок повідомлення завдовжки $n-1$ символів, то для побудови відрізка повідомлення завдовжки n попередній інтервал розбивається на стільки частин, скільки можливих значень має ДВВ. Для знаходження початку і кінця нового інтервалу повідомлення до початку попереднього інтервалу необхідно додати значення добутків його ширини на відповідні границі відрізка поточного нового символу з таблиці символів і їхніх інтервалів (таблиці кодера). З отриманих інтервалів вибирається той, що відповідає конкретному повідомленню завдовжки n символів. Для побудованого таким чином інтервалу повідомлення знаходиться число, що належить цьому відрізку, як правило, це ціле число, розділене на мінімальний степінь 2. Це дійсне число і буде кодом даного повідомлення. У міру

надходження символів повідомлення його інтервал звужується, відповідно кількість розрядів, необхідна для подання інтервалу збільшується. Більш імовірні символи меншою мірою звужують інтервал, ніж менш імовірні, і, отже, додають менше розрядів до результату.

Розв'язок задачі

Арифметичний алгоритм:

Побудуємо таблицю символів і відповідних їм інтервалів.

| Символ | Імовірність | Інтервал |
|--------|-------------|--------------|
| L | 1/10 | [0; 1/10) |
| A | 1/10 | [1/10; 2/10) |
| B | 1/10 | [2/10; 3/10) |
| O | 1/10 | [3/10; 4/10) |
| R | 1/10 | [4/10; 5/10) |
| E | 1/10 | [5/10; 6/10) |
| M | 1/10 | [6/10; 7/10) |
| U | 1/10 | [7/10; 8/10) |
| S | 1/10 | [8/10; 9/10) |
| ! | 1/10 | [9/10; 1) |

Ця таблиця зберігається разом із кодом стисненого повідомлення і призначена для кодування й декодування за арифметичним алгоритмом. Процес кодування повідомлення АВААААВ зручно подати у вигляді такої таблиці.

| Символ | Інтервал | Ширина |
|--------|----------------------------|-----------|
| L | [0,1/10) | 1/10 |
| A | [1/100; 2/100) | 1/100 |
| B | [12/1000; 13/1000) | 1/1000 |
| O | [123/10000; 124/10000) | 1/10000 |
| R | [1234/100000; 1235/100000) | 1/100000 |
| E | [12345/1000000; | 1/1000000 |

| | | |
|---|---|---------------|
| | 12346/1000000) | |
| M | [123456/10000000; 123457/10000000) | 1/10000000 |
| U | [1234567/100000000; 1234568/100000000) | 1/100000000 |
| S | [12345678/1000000000; 12345679/1000000000] | 1/1000000000 |
| ! | [123456789/10000000000; 12345679/1000000000) | 1/10000000000 |

Обчислення:

1) L – очевидно

2) A:

$$\text{Low} = 0 + 1/10 * 1/10 = 1/100$$

$$\text{High} = 0 + 1/10 * 2/10 = 2/100$$

3) B:

$$\text{Low} = 1/100 + 1/100 * 2/10 = 12/1000$$

$$\text{High} = 1/100 + 1/100 * 3/10 = 13/1000$$

4) O:

$$\text{Low} = 12/1000 + 1/1000 * 3/10 = 123/10000$$

$$\text{High} = 12/1000 + 1/1000 * 4/10 = 124/10000$$

5) R:

$$\text{Low} = 123/10000 + 1/10000 * 4/10 = 1234/100000$$

$$\text{High} = 123/10000 + 1/10000 * 5/10 = 1235/100000$$

6) E:

$$\text{Low} = 1234/100000 + 1/100000 * 5/10 = 12345/1000000$$

$$\text{High} = 1234/100000 + 1/100000 * 6/10 = 12346/1000000$$

7). M:

Low= $12345/1000000 + 1/1000000 * 6/10 = 123456/10000000$

High= $12345/1000000 + 1/1000000 * 7/10 = 123457/10000000$

8) U:

Low= $123456/10000000 + 1/10000000 * 7/10 = 1234567/100000000$

High= $123456/10000000 + 1/10000000 * 8/10 = 1234568/100000000$

9) S:

Low= $1234567/100000000 + 1/100000000 * 8/10 = 12345678/1000000000$

High= $1234567/100000000 + 1/100000000 * 9/10 = 12345679/1000000000$

10) !:

Low= $12345678/1000000000 + 1/1000000000 * 9/10 = 123456789/10000000000$

High= $12345678/1000000000 + 1/1000000000 * 1 = 12345679/1000000000$

Повідомлення **Laboremus!** однозначно визначає інтервал

$[123456789/1000000000; 12345679/1000000000) =$

$[0.0123456789; 0.012345679)$

Знайдемо дійсне число, що належить цьому інтервалу і є часткою від ділення цілого додатного числа на мінімальний степінь 2.

Таке число $424194297/2^{35} = 0,01234567890060134232044219970703$ належить інтервалу.

Двійкове подання чисельника буде арифметичним кодом повідомлення.

Розрядність коду визначається степенем 2. Отже, знайдемо двійковий

37-розрядний код числа :

$424194297_{10} = 00000011001010010001011000011111001_2$.

Таким чином, арифметичний код заданого повідомлення:

Code (**Laboremus!**)= $00000011001010010001011000011111001$.

Довжина коду $L(X)=35$ біт.

2. Обчислити ентропію джерела

$$H(X) = - \sum_{i=1}^k p_i \log_2 p_i = -\frac{10*1}{10} \log_2 \frac{1}{10} = -1*(-3,322) = 3.322 \left(\frac{\text{бит}}{\text{симв}} \right)$$

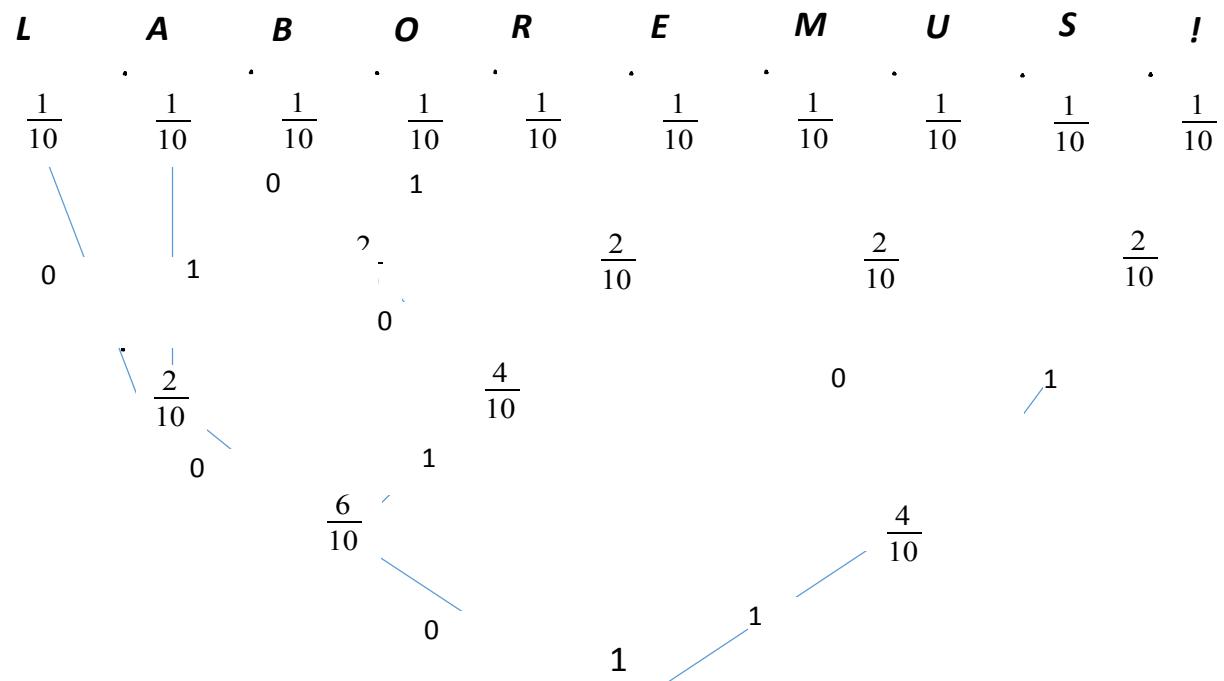
3. Виконати декодування закодованого за пунктом 1 повідомлення виразу.

Віднімаємо від поточного коду нижню границю інтервалу поточного символу і ділимо отриманий результат на ширину цього інтервалу і так з кожним разом.

4. $424194297/2^{35} = 0,01234567890060134232044219970703$
- “L”
5. $(0,01234567890060134232044219970703 - 0)/(1/10) =$
 $0,1234567890060134232044219970703 - "A"$
6. $(0,1234567890060134232044219970703 - 1/10)/(1/10) =$
 $0,234567890060134232044219970703 - "B"$
7. $(0,234567890060134232044219970703 - 2/10)/(1/10) =$
 $0,34567890060134232044219970703 - "O"$
8. $(0,34567890060134232044219970703 - 3/10)/(1/10) =$
 $0,4567890060134232044219970703 - "R"$
9. $(0,4567890060134232044219970703 - 4/10)/(1/10) =$
 $0,567890060134232044219970703 - "E"$
10. $(0,567890060134232044219970703 - 5/10)/(1/10) =$
 $0,67890060134232044219970703 - "M"$
11. $(0,67890060134232044219970703 - 6/10)/(1/10) =$
 $0,7890060134232044219970703 - "U"$
12. $(0,7890060134232044219970703 - 7/10)/(1/10) =$
 $0,890060134232044219970703 - "S"$
13. $(0,890060134232044219970703 - 8/10)/(1/10) =$
 $0,90060134232044219970703 - "!"$

Отже, отримали наше повідомлення «**Laboremus!**».

4. Побудувати дерево кодування виразу індивідуального завдання кодом Хаффмена.



| Буква, x_i | Імовірність p_i | Код, X_i | Довжина коду, l_i | $l_i p_i$ |
|--------------|-------------------|------------|---------------------|-----------|
| L | 1/10 | 000 | 3 | 3/10 |
| A | 1/10 | 001 | 3 | 3/10 |
| B | 1/10 | 0100 | 4 | 4/10 |
| O | 1/10 | 0101 | 4 | 4/10 |
| R | 1/10 | 0110 | 4 | 4/10 |
| E | 1/10 | 0111 | 4 | 4/10 |
| M | 1/10 | 100 | 3 | 3/10 |
| U | 1/10 | 101 | 3 | 3/10 |
| S | 1/10 | 110 | 3 | 3/10 |
| ! | 1/10 | 111 | 3 | 3/10 |

5. Закодуємо заданий вираз отриманим кодом:

| L | A | B | O | R | E | M | U | S | ! |
|-----|-----|------|------|------|------|-----|-----|-----|-----|
| 000 | 001 | 0100 | 0101 | 0110 | 0111 | 100 | 101 | 110 | 111 |

6. Обчислити довжину заданого за індивідуальним завданням виразу в ASCII кодах, за кодом Хаффмена і за арифметичним алгоритмом.

Довжина закодованого виразу кодом Хаффмена – 34 біт.

Довжина коду в ASCII – $8 \times 10 = 80$ (біт).

Довжина коду за арифметичним алгоритмом $L(X)=35$ біт.

Отже, найефективнішим виявився алгоритм Хаффмена (довжина коду – 34 біт), менш ефективний – арифметичний алгоритм (35 біт), найдовшим в даному випадку є кодування ASCII- кодами (80 біт).

Завдання

1. Для заданого коду і виразу за алгоритмом індивідуального завдання побудувати таблицю кодування (закодувати) повідомлення виразу. Для арифметичного алгоритму визначити таблицю розподілу символів.
2. Обчислити ентропію джерела.
3. Виконати декодування закодованого за пунктом 1 повідомлення виразу.
4. Побудувати дерево кодування виразу індивідуального завдання кодом Хаффмена.
5. Закодувати заданий вираз отриманим кодом Хаффмена
6. Обчислити довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим кодом (алгоритмом) і . кодом Хаффмена
7. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим кодом (алгоритмом) і . кодом Хаффмена. Зробити висновки
8. Оформити звіт з виконання етапів 1-7.

| Варіант | Завдання | Вираз | Переклад | Код |
|---------|----------|-------------------|--------------------|-----------------------|
| 13 | 62 | Theoria cum praxi | Теорія з практикою | Арифметичний алгоритм |

1. Для заданого коду і виразу за алгоритмом індивідуального завдання побудувати таблицю кодування (закодувати) повідомлення виразу. Для арифметичного алгоритму визначити таблицю розподілу символів

| Символ | Імовірність | Інтервал |
|--------|-------------|--------------|
| t | 1/17 | [0;1/17) |
| h | 1/17 | [1/17;2/17) |
| e | 1/17 | [2/17;3/17) |
| o | 1/17 | [3/17;4/17) |
| r | 2/17 | [4/17;6/17) |
| i | 2/17 | [6/17;8/17) |
| a | 2/17 | [8/17;10/17) |

| | | |
|---|------|---------------|
| c | 1/17 | [10/17;11/17) |
| u | 1/17 | [11/17;12/17) |
| m | 1/17 | [12/17;13/17) |
| p | 1/17 | [13/17;14/17) |
| x | 1/17 | [14/17;15/17) |
| - | 2/17 | [15/17;1) |

Межі інтервалів будемо шукати за наступними формулами:

$\text{High} = \text{Low}_{\text{old}} + (\text{High}_{\text{old}} - \text{Low}_{\text{old}}) * \text{Range}_{\text{High}}(x)$, $\text{Low} = \text{Low}_{\text{old}} + (\text{High}_{\text{old}} - \text{Low}_{\text{old}}) * \text{Range}_{\text{Low}}(x)$, где Low_{old} – нижня межа інтервалу, High_{old} – верхня межа інтервалу $\text{Range}_{\text{High}}$ и $\text{Range}_{\text{Low}}$ – верхня і нижня межі символу, що куdueться

Для першого символу ‘t’ кодер звужує інтервал до [0;1) до нового [0;1/17)

Наступний символ ‘h’

$$\begin{aligned} \text{low} &= 0 + 0.058823529411764705882352941176470588235294117647058823529 * \\ &0.058823529411764705882352941176470588235294117647058823529 = \\ &0.003460207612456747404844290657439446366782006920415224913 \end{aligned}$$

$$\begin{aligned} \text{height} &= 0 + 0.058823529411764705882352941176470588235294117647058823529 * \\ &0.117647058823529411764705882352941176470588235294117647058 = \\ &0.006920415224913494809688581314878892733564013840830449826 \end{aligned}$$

Наступний символ ‘e’

$$\begin{aligned} \text{low} &= 0.003460207612456747404844290657439446366782006920415224913 + \\ &0.003460207612456747404844290657439446366782006920415224913 * \\ &0.117647058823529411764705882352941176470588235294117647058 = \\ &0.003867290860981070628943618970079381233462243028699369020 \end{aligned}$$

$$\begin{aligned} \text{height} &= 0.003460207612456747404844290657439446366782006920415224913 + \\ &0.003460207612456747404844290657439446366782006920415224913 * \\ &0.176470588235294117647058823529411764705882352941176470588 = \\ &0.004070832485243232240993283126399348666802361082841441074 \end{aligned}$$

Наступний символ ‘o’

$$\begin{aligned} \text{low} &= 0.003867290860981070628943618970079381233462243028699369020 + \\ &0.000203541624262161612049664156319967433340118054142072053 * \\ &0.176470588235294117647058823529411764705882352941176470588 = \\ &0.003903209971144981501658265585900551956992852097077381735 \end{aligned}$$

$$\begin{aligned} \text{height} &= 0.003867290860981070628943618970079381233462243028699369020 + \\ &0.000203541624262161612049664156319967433340118054142072053 * \end{aligned}$$

```
0.235294117647058823529411764705882352941176470588235294117 =  
0.003915183007866285125896481124507608864836388453203385973
```

Наступний символ ‘r’

```
low = 0.003903209971144981501658265585900551956992852097077381735 +  
0.000011973036721303624238215538607056907843536356126004237 *  
0.235294117647058823529411764705882352941176470588235294117 =  
0.003906027156255876472067257477337506523544272416165853320
```

```
hight = 0.003903209971144981501658265585900551956992852097077381735 +  
0.000011973036721303624238215538607056907843536356126004237 *  
0.352941176470588235294117647058823529411764705882352941176 =  
0.003907435748811323957271753423055983806819982575710089112
```

Наступний символ ‘i’

```
low = 0.003906027156255876472067257477337506523544272416165853320 +  
1.408592555447485204495945718477283275710159544235792 × 10^-6 *  
0.352941176470588235294117647058823529411764705882352941176 =  
0.003906524306569563819786491340532263211759228943063818893
```

```
hight = 0.003906027156255876472067257477337506523544272416165853320 +  
1.408592555447485204495945718477283275710159544235792 × 10^-6 *  
0.470588235294117647058823529411764705882352941176470588235 =  
0.003906690023340792935692902628263848774497547785363140751
```

Наступний символ ‘a’

```
low = 0.003906524306569563819786491340532263211759228943063818893 +  
1.65716771229115906411287731585562738318842299321858 × 10^-7 *  
0.470588235294117647058823529411764705882352941176470588235 =  
0.003906602290932495168448331946523597594224320162969382120
```

```
hight = 0.003906524306569563819786491340532263211759228943063818893 +  
1.65716771229115906411287731585562738318842299321858 × 10^-7 *  
0.588235294117647058823529411764705882352941176470588235294 =  
0.003906621787023228005613792098021431189840592967945772927
```

Наступний символ ‘_’

```
low = 0.003906602290932495168448331946523597594224320162969382120 +  
1.9496090732837165460151497833595616272804976390807 × 10^-8 *  
0.882352941176470588235294117647058823529411764705882352941 =  
0.003906619493365494730653149727256980178591619696772079890
```

```
hight = 0.003906602290932495168448331946523597594224320162969382120 +  
1.9496090732837165460151497833595616272804976390807 × 10^-8 * 1 =  
0.003906621787023228005613792098021431189840592967945772927
```

Наступний символ ‘c’

```
low = 0.003906619493365494730653149727256980178591619696772079890 +
2.293657733274960642370764451011248973271173693037 × 10^-9 *
0.588235294117647058823529411764705882352941176470588235294 =
0.003906620842575926068865292298294892538149839268050722852
```

```
hight = 0.003906619493365494730653149727256980178591619696772079890 +
2.293657733274960642370764451011248973271173693037 × 10^-9 *
0.647058823529411764705882352941176470588235294117647058823 =
0.003906620977496969202686506555398683774105661225178587149
```

Наступний символ ‘u’

```
low = 0.003906620842575926068865292298294892538149839268050722852 +
1.34921043133821214257103791235955821957127864297 × 10^-10 *
0.647058823529411764705882352941176470588235294117647058823 =
0.00390662092987777508396666229362051573180077005015811514
```

```
hight = 0.003906620842575926068865292298294892538149839268050722852 +
1.34921043133821214257103791235955821957127864297 × 10^-10 *
0.705882352941176470588235294117647058823529411764705882352 =
0.003906620937814309457444972950368156940001007708376274120
```

Наступний символ ‘m’

```
low = 0.003906620937814309457444972950368156940001007708376274120 +
7.936531949048306721006105366820930703360462606 × 10^-12 *
0.705882352941176470588235294117647058823529411764705882352 =
0.003906620943416567303832012988725407787168723498983659488
```

```
hight = 0.003906620937814309457444972950368156940001007708376274120 +
7.936531949048306721006105366820930703360462606 × 10^-12 *
0.764705882352941176470588235294117647058823529411764705882 =
0.003906620943883422124364266325255178691099366481534274936
```

Наступний символ ‘_’

```
low = 0.003906620943416567303832012988725407787168723498983659488 +
4.66854820532253336529770903930642982550615448 × 10^-13 *
0.882352941176470588235294117647058823529411764705882352941 =
0.003906620943828498027831060050369323290636937895351849589
```

```
hight = 0.003906620943416567303832012988725407787168723498983659488 +
4.66854820532253336529770903930642982550615448 × 10^-13 * 1 =
0.003906620943883422124364266325255178691099366481534274936
```

Наступний символ ‘р’

```
low = 0.003906620943828498027831060050369323290636937895351849589 +
5.4924096533206274885855400462428586182425347 × 10^-14 *
0.764705882352941176470588235294117647058823529411764705882 =
0.003906620943870498807532923672340859773343500931844292501
```

```

high = 0.003906620943828498027831060050369323290636937895351849589 +
5.4924096533206274885855400462428586182425347 × 10^-14 *
0.823529411764705882352941176470588235294117647058823529411 =
0.003906620943873729636740759335569439502782467319266788110

```

Наступний символ ‘r’

```

low = 0.003906620943870498807532923672340859773343500931844292501 +
3.230829207835663228579729438966387422495609 × 10^-15 *
0.235294117647058823529411764705882352941176470588235294117 =
0.003906620943871259002640649710747584415564434199473114997

```

```

high = 0.003906620943870498807532923672340859773343500931844292501 +
3.230829207835663228579729438966387422495609 × 10^-15 *
0.352941176470588235294117647058823529411764705882352941176 =
0.003906620943871639100194512729950946736674900833287526245

```

Наступний символ ‘a’

```

low = 0.003906620943871259002640649710747584415564434199473114997 +
3.8009755386301920336232110466633814411248 × 10^-16 *
0.470588235294117647058823529411764705882352941176470588235 =
0.003906620943871437872077761719784460801969359674209308525

```

```

high = 0.003906620943871259002640649710747584415564434199473114997 +
3.8009755386301920336232110466633814411248 × 10^-16 *
0.588235294117647058823529411764705882352941176470588235294 =
0.003906620943871482589437039722043679898570591042893356907

```

Наступний символ ‘x’

```

low = 0.003906620943871437872077761719784460801969359674209308525 +
4.4717359278002259219096601231368684048382 × 10^-17 *
0.823529411764705882352941176470588235294117647058823529411 =
0.00390662094387147469813834360399793535211550213125583663

```

```

high = 0.003906620943871437872077761719784460801969359674209308525 +
4.4717359278002259219096601231368684048382 × 10^-17 *
0.882352941176470588235294117647058823529411764705882352941 =
0.003906620943871477328571242310013183534264563823048174744

```

Наступний символ ‘i’

```

low = 0.00390662094387147469813834360399793535211550213125583663 +
2.630432898706015248182153013609922591081 × 10^-18 *
0.352941176470588235294117647058823529411764705882352941176 =
0.003906620943871475626526425500238611181106731487215909926

```

0.003906620943871475935989119465652169790771791911912685348

```

high = 0.00390662094387147469813834360399793535211550213125583663 +
2.630432898706015248182153013609922591081 × 10^-18 *

```

$0.47058823529411764705882352941176470588235 =$
 $0.003906620943871475935989119465652169790771791911912685348$

Послідовність кодування інтервалів повідомлення наведена в таблиці

| Символ-інтервал | Інтервал повідомлення | Ширина інтервалу |
|-------------------|---|--|
| $t - [0;1/17)$ | $[0;0.05882352941176470588235294117647058823529]$ | $0.05882352941176470588235294117647058823529$ |
| $h - [1/17;2/17)$ | $[0.0034602076124567474048;0.0069204152249139688581314878892733564013840830449826)$ | $0.003460207612456747404844290657439446366782006920415224913$ |
| $e - [2/17;3/17)$ | $[0.003867290860981070628943618970079381233462243028699369020;0.004070832485243232240993283126399348666802361082841441074)$ | $0.000203541624262161612049664156319967433340118054142072053$ |
| $o - [3/17;4/17)$ | $[0.003903209971144981501658265585900551956992852097077381735;0.003915183007866285125896481124507608864836388453203385973)$ | $0.000011973036721303624238215538607056907843536356126004237$ |
| $r - [4/17;6/17)$ | $[0.003906027156255876472067257477337506523544272416165853320;0.003907435748811323957271753423055983806819982575710089112)$ | $1.408592555447485204495945718477283275710159544235792 \times 10^{-6}$ |
| $i - [6/17;8/17)$ | $[0.003906524306569563819786491340532263211759228943063818893;0.003906690023340792935692902628263848774497547785363140751)$ | $1.65716771229115906411287731585562738318842299321858 \times 10^{-7}$ |

| | | |
|-------------------|---|--|
| a - [8/17;10/17) | [0.00390660229093249516 8448331946523597594224 320162969382120; 0.003906621787023228005 6137920980214311898405 92967945772927) | 1.9496090732837165460151 49783359561627280497639 0807×10^{-8} |
| _ - [15/17;1) | [0.00390661949336549473 0653149727256980178591 619696772079890; 0.003906621787023228005 6137920980214311898405 92967945772927) | 2.2936577332749606423707 64451011248973271173693 037×10^{-9} |
| c - [10/17;11/17) | [0.00390662084257592606 8865292298294892538149 839268050722852; 0.003906620977496969202 6865065553986837741056 61225178587149) | 1.3492104313382121425710 37912359558219571278642 97×10^{-10} |
| u - [11/17;12/17) | [0.00390662092987777750 8396666229362051573180 077005015811514; 0.003906620937814309457 4449729503681569400010 07708376274120) | 7.9365319490483067210061 05366820930703360462606 $\times 10^{-12}$ |
| m - [12/17;13/17) | [0.00390662094341656730 3832012988725407787168 723498983659488; 0.003906620943883422124 3642663252551786910993 66481534274936) | 4.6685482053225333652977 0903930642982550615448 $\times 10^{-13}$ |
| _ - [15/17;1) | [0.00390662094382849802 7831060050369323290636 937895351849589; 0.003906620943883422124 3642663252551786910993 66481534274936) | 5.4924096533206274885855 $400462428586182425347 \times$ 10^{-14} |
| p - [13/17;14/17) | [0.00390662094387049880 7532923672340859773343 500931844292501; 0.003906620943873729636 7407593355694395027824 67319266788110) | 3.2308292078356632285797 $29438966387422495609 \times$ 10^{-15} |

| | | |
|-------------------|---|--|
| r - [4/17;6/17) | [0.00390662094387125900 2640649710747584415564 434199473114997; 0.003906620943871639100 1945127299509467366749 00833287526245) | 3.8009755386301920336232 $1110466633814411248 \times$ 10^{-16} |
| a - [8/17;10/17) | [0.00390662094387143787 2077761719784460801969 359674209308525; 0.003906620943871482589 4370397220436798985705 91042893356907) | 4.4717359278002259219096 $601231368684048382 \times$ 10^{-17} |
| x - [14/17;15/17) | [0.00390662094387147469 8138343603997935352111 550213125583663; 0.003906620943871477328 5712423100131835342645 63823048174744) | 2.6304328987060152481821 $53013609922591081 \times 10^{-18}$ |
| i - [6/17;8/17) | [0.00390662094387147562 6526425500238611181106 731487215909926; 0.003906620943871475935 9891194656521697907717 91911912685348) | 3.0946269396541355860966 $5060424696775422 \times 10^{-19}$ |

З результату кодування повідомлення «theoria_cum_praxi» отримаємо:
 $[0.003906620943871475626526425500238611181106731487215909926;$
 $0.003906620943871475935989119465652169790771791911912685348)$

Для цього числа знаходять мінімальний степінь:
(Степінь = (Кількість співпадаючих розрядів / 2) * 7 = (18 / 2) * 7 = 63)
 $0.003906620943871475781257772482945390485939261699564297636 =$
 $36032218372295380 / 2^{63}$ Вона визначає 63 - розрядний код повідомлення.
Двійковий 63-розрядний код числа 36032218372295380 =
10000000000000110001110010011000100011001100101011010000
арифметичним кодом даного повідомлення.

2. Обчислити ентропію джерела

$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i = -\left(\frac{9}{17} * \log_2 \frac{1}{17} + \frac{8}{17} * \log_2 \frac{2}{17}\right) = -(0.529411 * (-4.087)) +$$

$$0.470588*(-3.087)) = 3.616407913$$

3. Виконати декодування закодованого за пунктом 1 повідомлення виразу

Декодування арифметичного коду повідомлення здійснюється за таким алгоритмом. Декодеру і кодеру відома таблиця розподілу інтервалів символів алфавіту.

Крок 1 За таблицею інтервалів символів алфавіту визначають відрізок, що містить значення поточного коду, і за ним однозначно визначають символ повідомлення. Якщо символ - маркер кінця повідомлення, то кінець, інакше - перейти до кроку 2.

Крок 2 Від поточного коду віднімається нижня границя його інтервалу.

Різниця ділиться на довжину цього інтервалу. Отримане значення вважається новим значенням поточного коду. Перейти до кроку 1

| Символ | Імовірність | Інтервал |
|--------|-------------|---------------------|
| t | 1/17 | [0;0.058823) |
| h | 1/17 | [0.058823;0.117647) |
| е | 1/17 | [0.117647;0.176470) |
| o | 1/17 | [0.176470;0.235294) |
| г | 2/17 | [0.235294;0.352941) |
| і | 2/17 | [0.352941;0.470588) |
| а | 2/17 | [0.470588;0.588235) |
| с | 1/17 | [0.588235;0.647058) |
| у | 1/17 | [0.647058;0.705882) |
| m | 1/17 | [0.705882;0.764706) |
| p | 1/17 | [0.764706;0.823529) |
| x | 1/17 | [0.823529;0.882353) |
| - | 2/17 | [0.882353;1) |

Дійсне число, яке належить інтервалу, що однозначно визначає закодоване повідомлення 0.003906620943871475781257772482945390485939261699564297636 є значенням поточного коду. З таблиці наведеної вище визначають відрізок [0;0.058823), якому належить це число і перший закодований символ 't'. Віднімаємо від поточного коду нижню границю інтервалу символу 't' і ділимо отриманий результат на ширину цього інтервалу.

$0.003906620943871475781257772482945390485939261699564297636 / 0.0588235294117647058$
 $82352941176470588235294117647058823529 = 0.06641255604581508554$
 Це число належить інтервалу $[0.058823; 0.117647)$, отже це символ ‘h’

$(0.06641255604581508554 -$
 $0.05882352941176470588235294117647058823529) /$
 $0.05882352941176470588235294117647058823529 =$
 0.12901345277885645418
 Це число належить інтервалу $[0.117647; 0.176470)$, отже це символ ‘e’

$(0.12901345277885645418 - 0.1176470588235294117647058823) /$
 $0.05882352941176470588235294117647058823529 =$
 0.19322869724055972106
 Це число належить інтервалу $[0.176470; 0.235294)$, отже це символ ‘o’

$(0.19322869724055972106 -$
 $0.17647058823529411764705882352941176470588) /$
 $0.05882352941176470588235294117647058823529 =$
 0.28488785308951525802
 Це число належить інтервалу $[0.235294; 0.352941)$, отже це символ ‘r’

$(0.28488785308951525802 -$
 $0.235294117647058823529411764705882352941176470588235294117) /$
 $0.117647058823529411764705882352941176470588235294117647058 =$
 0.42154675126087969317
 Це число належить інтервалу $[0.352941; 0.470588)$, отже це символ ‘i’

$(0.42154675126087969317 -$
 $0.352941176470588235294117647058823529411764705882352941176) /$
 $0.117647058823529411764705882352941176470588235294117647058 =$
 0.583147385717477391945
 Це число належить інтервалу $[0.470588; 0.588235)$, отже це символ ‘a’

$(0.583147385717477391945 -$
 $0.470588235294117647058823529411764705882352941176470588235) /$
 $0.117647058823529411764705882352941176470588235294117647058 =$
 0.9567527785985578315325
 Це число належить інтервалу $[0.882353; 1)$, отже це символ ‘_’

$(0.9567527785985578315325 -$
 $0.882352941176470588235294117647058823529411764705882352941) /$
 $0.117647058823529411764705882352941176470588235294117647058 =$
 $0.63239861808774156802625$
 Це число належить інтервалу $[0.588235; 0.647058)$, отже це символ ‘c’

$(0.63239861808774156802625 -$
 $0.588235294117647058823529411764705882352941176470588235294) /$
 $0.058823529411764705882352941176470588235294117647058823529 =$
 $0.691952978079841950563897058823529411764705882352941176469$

Це число належить інтервалу [0.647058;0.705882), отже це символ ‘и’

$$(0.691952978079841950563897058823529411764705882352941176469 - 0.647058823529411764705882352941176470588235294117647058823) / 0.058823529411764705882352941176470588235294117647058823529 = 0.76320062735731315958625$$

Це число належить інтервалу [0.705882;0.764706), отже це символ ‘м’

$$(0.76320062735731315958625 - 0.70588235294117647058823529411764705882352) / 0.058823529411764705882352941176470588235294117647058823529 = 0.97441066507432371296625$$

Це число належить інтервалу [0.882353;1), отже це символ ‘_’

$$(0.97441066507432371296625 - 0.88235294117647058823529411764705882352941) / 0.117647058823529411764705882352941176470588235294117647058 = 0.782490653131751560213125$$

Це число належить інтервалу [0.764706;0.823529), отже це символ ‘р’

$$(0.782490653131751560213125 - 0.764705882352941176470588235294117647058823529411764705882) / 0.058823529411764705882352941176470588235294117647058823529 = 0.302341103239776523623125$$

Це число належить інтервалу [0.235294;0.352941), отже це символ ‘г’

$$(0.302341103239776523623125 - 0.235294117647058823529411764705882352941176470588235294117) / 0.117647058823529411764705882352941176470588235294117647058 = 0.5698993775381004507965625$$

Це число належить інтервалу [0.470588;0.588235), отже це символ ‘а’

$$(0.5698993775381004507965625 - 0.470588235294117647058823529411764705882352941176470588235) / 0.117647058823529411764705882352941176470588235294117647058 = 0.84414470907385383177078125$$

Це число належить інтервалу [0.823529;0.882353), отже це символ ‘х’

$$(0.84414470907385383177078125 - 0.823529411764705882352941176470588235294117647058823529411) / 0.058823529411764705882352941176470588235294117647058823529 = 0.37046005425551514010328125$$

Це число належить інтервалу [0.352941;0.470588), отже це символ ‘і’

10. Постановка задачі

1. Для заданого коду і виразу за алгоритмом індивідуального завдання побудувати таблицю кодування (закодувати) повідомлення виразу. Для арифметичного алгоритму визначити таблицю розподілу символів.
2. Обчислити ентропію джерела.
3. Виконати декодування закодованого за пунктом 1 повідомлення виразу.
4. Побудувати дерево кодування виразу індивідуального завдання кодом Хаффмена.
5. Закодувати заданий вираз отриманим кодом Хаффмена
6. Обчислити довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим кодом (алгоритмом) і кодом Хаффмена
7. Порівняти довжини заданого за індивідуальним завданням виразу в ASCII-кодах, закодованого виразу за заданим кодом (алгоритмом) і кодом Хаффмена. Зробити висновки

| Варіант | Завдання | Вираз | Переклад | Код |
|----------------|-----------------|--------------------|--|-----------------------|
| 3. | 88 | Si sapis, sis apis | Якщо ти розумний, будь (працьовитим), як бджола. | Арифметичний алгоритм |

11. Розв'язок задачі

2.1 Для заданого коду і виразу за алгоритмом індивідуального завдання побудувати таблицю кодування (закодувати) повідомлення виразу. Для арифметичного алгоритму визначити таблицю розподілу символів.

Заданий вираз (si sapis, sis apis) складається з 18 символів, визначимо скільки разів повторюється кожен окремий символ в данному виразі:

Табл.1

| символ | s | i | □ | a | p | , |
|--------------------|---|---|---|---|---|---|
| Кількість повторів | 6 | 4 | 3 | 2 | 2 | 1 |

За допомогою Табл.1 складемо таблицю розподілу символів

$$(P(X=x_i) = \frac{n x_i (\text{кількість повторів символа } x_i)}{n(\text{загальна кількість символів у виразі}})$$

Табл.2

| X_i | s | i | □ | a | p | , |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|
| P_i | $\frac{6}{18}$ | $\frac{4}{18}$ | $\frac{3}{18}$ | $\frac{2}{18}$ | $\frac{2}{18}$ | $\frac{1}{18}$ |

Перевіримо умови нормування:

$$\sum_i P(X=x_i)=1$$

$$\sum_i^{20} P(X=x_i)=\frac{6}{18}+\frac{4}{18}+\frac{3}{18}+\frac{2}{18}+\frac{2}{18}+\frac{1}{18}=1$$

a. Обчислити ентропію джерела.

Знайдемо ентропію джерела X:

$$H(X)=-\sum_i p_i \log_2 p_i$$

$$H(X)=-\left(\frac{6}{18} \log_2 \frac{6}{18} + \frac{4}{18} \log_2 \frac{4}{18} + \frac{3}{18} \log_2 \frac{3}{18} + \frac{2}{18} \log_2 \frac{2}{18} + \frac{2}{18} \log_2 \frac{2}{18} + \frac{1}{18} \log_2 \frac{1}{18}\right) \approx -\left(\frac{6}{18} (-1,585) + \frac{4}{18} (-2,17) + \frac{3}{18} (-2,585) + \frac{2*2}{18} (-3,17) + \frac{1}{18} (-4,17)\right) \approx 2.3775 \text{ (біт/символ)}$$

Алгоритм арифметичного кодування

Спочатку за розподілом ймовірностей ДВВ складається таблиця з пересічних в граничних точках відрізків для кожного із значень ДВВ

Об'єднання цих відрізків утворює інтервал $[0;1]$, а їхні довжини пропорційні ймовірностям значень ДВВ.

Алгоритм кодування полягає в побудові інтервалу, що однозначно визначає конкретну послідовність значень ДВВ у такий спосіб. З відрізу повідомлення завдовжки $n-1$ символів для побудови відрізу повідомлення завдовжки n попередній інтервал розбивається на стільки частин, скільки можливих значень має ДВВ. Для знаходження початку і кінця нового інтервалу повідомлення до початку попереднього інтервалу необхідно додати значення добутків його ширини на відповідні граници відрізу поточного нового символу з таблиці символів і їхніх інтервалів (таблиці кодера). З отриманих інтервалів вибирається той, що відповідає конкретному повідомленню завдовжки n символів.

Для побудованого таким чином інтервалу повідомлення знаходиться число, що належить цьому відрізку. Зазвичай це ціле число, розділене на мінімальний степінь 2. Це дійсне число і буде кодом даного повідомлення.

За надходженням символів повідомлення його інтервал звужується, а відповідна кількість розрядів, необхідна для подання інтервалу, збільшується.

Більш імовірні символи меншою мірою звужують інтервал, ніж менш імовірні, і, отже, додають менше розрядів до результату.

Частоти символів і відрізки символів виразу «si sapis, sis apis» подані таблиці 3.

Таблиця 3— Імовірності появи і інтервали символів

| <i>Символ</i> | <i>Імовірність</i> | <i>Інтервал</i> |
|---------------|--------------------|-----------------------|
| <i>s</i> | 6/18 | <i>[0; 6/18)</i> |
| <i>i</i> | 4/18 | <i>[6/18; 10/18)</i> |
| \square | 3/18 | <i>[10/18; 13/18)</i> |
| <i>a</i> | 2/18 | <i>[13/18; 15/18)</i> |
| <i>p</i> | 2/18 | <i>[15/18; 17/18)</i> |
| , | 1/18 | <i>[17/18; 1,0)</i> |

Для першого символу «s» кодер звужує початковий інтервал $[0; 1)$ до нового $[0; 0.3333]$.

Наступний символ «і» кодується підінтервалом $[0.11108889; 0.18518148]$:
 $\text{lowi}=0+0.3333*0.3333=0.11108889$; $\text{highi}=0+0.3333*0.5556=0.18518148$.

Наступний символ «□» кодується підінтервалом

$[0.152254733004; 0.164598558498]$:

$\text{lowi}=0.11108889+0.07409259*0.5556=0.152254733004$;

$\text{highi}=0.11108889+0.07409259*0.7222=0.164598558498$.

Наступний символ «s» кодується підінтервалом

$[0.152254733004; 0.1563689300411502]$:

$\text{lowi}=0.152254733004+0.012343825494*0=0.152254733004$;

$\text{highi}=0.152254733004+0.012343825494*0.3333=0.1563689300411502$.

Наступний символ «а» кодується підінтервалом

$[0.15522600610422987444; 0.15568309339505726166]$:

$\text{lowi}=0.152254733004+0.0041141970371502*0.7222=0.15522600610422987444$;

$\text{highi}=0.152254733004+0.0041141970371502*0.8333=0.15568309339505726166$.

Наступний символ «р» кодується підінтервалом

$[0.155606896943676336210426; 0.155657679341687258930568]$:

$\text{lowi}=0.15522600610422987444+0.00045708729082738722*0.8333=$

$=0.155606896943676336210426$;

$\text{highi}=0.15522600610422987444+0.00045708729082738722*0.9444=$

$=0.155657679341687258930568$.

Наступний символ «і» кодується підінтервалом

$[0.1556238227169333767530493286; 0.1556351116440112048737368952]$:

$\text{lowi}=0.155606896943676336210426+0.000050782398010922720142*0.3333=$

$=0.1556238227169333767530493286$;

$\text{highi}=0.155606896943676336210426+0.000050782398010922720142*0.5556=$

$=0.1556351116440112048737368952$.

Наступний символ «s» кодується підінтервалом

$[0.1556238227169333767530493286; 0.15562758531632841686567449454778]$:

$\text{lowi}=0.1556238227169333767530493286+0.0000112889270778281206875666*0=$

$=0.1556238227169333767530493286$;

$\text{highi}=0.1556238227169333767530493286+0.0000112889270778281206875666*$

$*0.3333=0.15562758531632841686567449454778$.

Наступний символ «,» кодується підінтервалом

$[0.155627376115802052635412535321083432$;

$0.15562758531632841686567449454778]$:

$\text{lowi}=0.1556238227169333767530493286+0.0000037625993950401126251659477$

$8*0.9444=0.155627376115802052635412535321083432$;

$\text{highi}=0.1556238227169333767530493286+0.000003762599395040112625165947$

$8*1=0.15562758531632841686567449454778$.

Наступний символ «□» кодується підінтервалом
 $[0.1556274923476145006017460798674360451808;$
 $0.1556275272004221928825077222746036934096]:$
 $\text{lowi}=0.155627376115802052635412535321083432+0.00000020920052636423026$
 $1959226696568*0.5556=0.1556274923476145006017460798674360451808;$
 $\text{highi}=0.155627376115802052635412535321083432+0.0000002092005263642302$
 $61959226696568*0.7222=0.1556275272004221928825077222746036934096).$

Наступний символ «s» кодується підінтервалом
 $[0.1556274923476145006017460798674360451808;$
 $0.15562750396405530443892393528174502233545904]:$
 $\text{lowi}=0.1556274923476145006017460798674360451808+0.0000000348528076922$
 $807616424071676482288*0=0.1556274923476145006017460798674360451808;$
 $\text{highi}=0.1556274923476145006017460798674360451808+0.000000034852807692$
 $2807616424071676482288*0.3333=0.1556275039640553044389239352817450223$
 $3545904).$

Наступний символ «i» кодується підінтервалом
 $[0.155627496219374220520677459077025227266447858032;$
 $0.155627498801709011213682096335626112887928562624]:$
 $\text{lowi}=0.1556274923476145006017460798674360451808+0.0000000116164408038$
 $3717785541430897715465904*0.3333=0.1556274962193742205206774590770252$
 $27266447858032;$
 $\text{highi}=0.1556274923476145006017460798674360451808+0.000000011616440803$
 $83717785541430897715465904*0.5556=0.155627498801709011213682096335626$
 112887928562624

Наступний символ «s» кодується підінтервалом
 $[0.155627496219374220520677459077025227266447858032;$
 $0.1556274970800664062586559046753169024440873768725136]:$
 $\text{lowi}=0.155627496219374220520677459077025227266447858032+0.00000000258$
 $2334790693004637258600885621480704592*0=0.15562749621937422052067745$
 $9077025227266447858032;$
 $\text{highi}=0.155627496219374220520677459077025227266447858032+0.0000000025$
 $82334790693004637258600885621480704592*0.3333=0.155627497080066406258$
 $6559046753169024440873768725136).$

Наступний символ «□» кодується підінтервалом
 $[0.15562749669757479891669828345143608199514437469978935616;$
 $0.15562749684096611706064549248811147507973911853861892192]:$

$\text{lowi} = 0.155627496219374220520677459077025227266447858032 + 0.000000000086$
 $06921857379784455982916751776395188405136 * 0.5556 = 0.155627496697574798$
 $91669828345143608199514437469978935616;$
 $\text{highi} = 0.155627496219374220520677459077025227266447858032 + 0.00000000008$
 $606921857379784455982916751776395188405136 * 0.7222 = 0.15562749684096611$
 $706064549248811147507973911853861892192).$

Наступний символ «а» кодується підінтервалом
 $[0.155627496801132008880256957817723050880838698700192068551872;$
 $0.155627496817062784326049492741697687052537174740686033307808):$
 $\text{lowi} = 0.15562749669757479891669828345143608199514437469978935616 + 0.000$
 $00000014339131814394720903667539308459474383882956576 * 0.7222 = 0.155627$
 $496801132008880256957817723050880838698700192068551872;$
 $\text{highi} = 0.15562749669757479891669828345143608199514437469978935616 + 0.00$
 $000000014339131814394720903667539308459474383882956576 * 0.8333 = 0.15562$
 $7496817062784326049492741697687052537174740686033307808).$

Наступний символ «р» кодується підінтервалом
 $[0.1556274968144071240592358771698711152027150387847356893829934688;$
 $0.1556274968161770332112634277999246972813907394728345688673779584):$
 $\text{lowi} = 0.155627496801132008880256957817723050880838698700192068551872 +$
 $0.000000000015930775445792534923974636171698476040493964755936 * 0.8333$
 $= 0.1556274968144071240592358771698711152027150387847356893829934688;$
 $\text{highi} = 0.155627496801132008880256957817723050880838698700192068551872 +$
 $0.000000000015930775445792534923974636171698476040493964755936 * 0.9444$
 $= 0.1556274968161770332112634277999246972813907394728345688673779584).$

Наступний символ «і» кодується підінтервалом
 $[0.1556274968149970347796066597948679741095376498240790459151388191;$
 $0.1556274968153904855841023842999288854056272580870434268245174912):$
 $\text{lowi} = 0.15562749681440712405923587716987111520271503878473568938299346$
 $88 + 0.000000000001769909152027550630053582078675700688098879484384489$
 $6 * 0.3333 = 0.1556274968149970347796066597948679741095376498240790459151$
 $388191;$
 $\text{highi} = 0.1556274968144071240592358771698711152027150387847356893829934$
 $688 + 0.00000000000176990915202755063005358207867570068809887948438448$
 $96 * 0.5556 = 0.155627496815390485584102384299928885405627258087043426824$
 $5174912).$

Наступний символ «s» кодується підінтервалом
 $[0.1556274968149970347796066597948679741095376498240790459151388191;$
 $0.1556274968151281719327450847724047758445243162581250740722347305):$

lowi=0.15562749681499703477960665979486797410953764982407904591513881
 91+0.000000000000393450804495724505060911296089608262964380909378672
 1*0=0.15562749681499703477960665979486797410953764982407904591513881
 9;
 highi=0.1556274968149970347796066597948679741095376498240790459151388
 191+0.00000000000039345080449572450506091129608960826296438090937867
 21*0.3333=0.155627496815128171932745084772404775844524316258125074072
 2347305).

Послідовність інтервалів кодування виразу «si sapis, sis apis» наведена в таблиці 4.

Таблиця 4 – Кодування виразу

| <i>Символ – інтервал</i> | <i>Інтервал повідомлення</i> | <i>Ширина інтервалу</i> |
|---------------------------|---|--|
| s - [0; 6/18) | [0; 0.3333) | 0,3333 |
| i - [6/18; 10/18) | [0.11108889; 0.18518148) | 0.07409259 |
| _ - [10/18; 13/18) | [0.152254733004; 0.164598558498) | 0.012343825494 |
| s - [0; 6/18) | [0.152254733004; 0.1563689300411502) | 0.0041141970371502 |
| a - [13/18; 15/18) | [0.15522600610422987444; 0.15568309339505726166) | 0.00045708729082738722 |
| p - [15/18; 17/18) | [0.155606896943676336210426; 0.155657679341687258930568) | 0.000050782398010922720142 |
| i - [6/18; 10/18) | [0.1556238227169333767530493286; 0.155635116440112048737368952) | 0.0000112889270778281206875666 |
| s - [0; 6/18) | [0.1556238227169333767530493286; 0.15562758531632841686567449454778) | 0.376259939504011262516594778 10^{-5} |
| , - [17/18; 1,0) | [0.155627376115802052635412535321083432; 0.15562758531632841686567449454778) | 0.2092005263642302619592266965 $68 * 10^{-6}$ |
| _ - [10/18; 13/18) | [0.1556274923476145006017460798674360451808; 0.1556275272004221928825077222746036934096) | 0.3485280769228076164240716764 $82288 * 10^{-7}$ |
| s-[0; 6/18) | [0.1556274923476145006017460798674360451808; 0.15562750396405530443892393528174502233545 904) | 0.1161644080383717785541430897 $715465904 * 10^{-7}$ |
| i-[6/18; 10/18) | [0.1556274962193742205206774590770252272664 47858032] | 0.2582334790693004637258600885 |

| | | |
|--------------------------|---|--|
| | [0.15562749880170901121368209633562611288792 8562624) | 621480704592 * 10⁻⁸ |
| s-[0; 6/18) | [0.1556274962193742205206774590770252272664 47858032; 0.15562749708006640625865590467531690244408 73768725136) | 0.8606921857379784455982916751 776395188405136 * 10⁻⁹ |
| _ -[10/18; 13/18) | [0.1556274966975747989166982834514360819951 4437469978935616; 0.15562749684096611706064549248811147507973 911853861892192) | 0.1433913181439472090366753930 8459474383882956576 * 10⁻⁹ |
| a-[13/18; 15/18) | [0.1556274968011320088802569578177230508808386987 00192068551872; 0.15562749681706278432604949274169768705253717474 068603307808) | 0.1593077544579253492397463617 1698476040493964755936 * 10⁻¹⁰ |
| p-[15/18; 17/18) | [0.1556274968144071240592358771698711152027150387 847356893829934688; 0.15562749681617703321126342779992469728139073947 28345688673779584) | 0.1769909152027550630053582078 6757006880988794843844896 * 10⁻¹¹ |
| i-[6/18; 10/18) | [0.1556274968149970347796066597948679741095376498 240790459151388191; 0.15562749681539048558410238429992888540562725808 70434268245174912) | 0.3934508044957245050609112960 896082629643809093786721 * 10⁻¹¹ |
| s-[0; 6/18) | [0.1556274968149970347796066597948679741095 376498240790459151388191; 0.15562749681512817193274508477240477584452 43162581250740722347305) | 0.1311371531384249775368017349 866664340460281570959114 * 10⁻¹² |

З результату кодування виразу «si sapis, sis apis» отримаємо інтервал $[0.1556274968149970347796066597948679741095376498240790459151388191;$ $0.1556274968151281719327450847724047758445243162581250740722347305]$.

Для цього числа знаходять мінімальний

степінь 2 - це $0,1556274968151 = 85557121175 / 2^{39}$ визначає 39 - розрядний код повідомлення. Двійковий 39-розрядний код числа $85557121175 = 001001111101011100110100001000010010111$ є арифметичним кодом даного повідомлення:

Code(si sapis, sis apis)= **001001111101011100110100001000010010111.**

- b. Виконати декодування закодованого за пунктом 1 повідомлення виразу.

Декодування арифметичного коду повідомлення здійснюється за таким алгоритмом. Декодеру і кодеру відома таблиця розподілу інтервалів символів алфавіту.

Крок 1 За таблицею інтервалів символів алфавіту визначають відрізок, що містить значення поточного коду, і за ним однозначно визначають символ повідомлення. Якщо символ - маркер кінця повідомлення, то кінець, інакше - пейти до кроку 2.

Крок 2 Від поточного коду віднімається нижня границя його інтервалу. Різниця ділиться на довжину цього інтервалу. Отримане значення вважається новим значенням поточного коду. Перейти до кроку 1.

Арифметичний код повідомлення з 18 символів є

$001001111101011100110100001000010010111_2 = 85557121175_{10}$, а символи і

інтервали повідомлення наведені у таблиці 3. Декодувати це повідомлення.

Таблиця 3 – Імовірності появи і інтервали символів

| Символ | Імовірність | Інтервал |
|--------|-------------|-----------------------|
| s | 6/18 | <i>[0; 6/18)</i> |
| i | 4/18 | <i>[6/18; 10/18)</i> |
| □ | 3/18 | <i>[10/18; 13/18)</i> |
| a | 2/18 | <i>[13/18; 15/18)</i> |
| p | 2/18 | <i>[15/18; 17/18)</i> |
| , | 1/18 | <i>[17/18; 1,0)</i> |

Дійсне число, яке належить інтервалу, що однозначно визначає закодоване повідомлення $0,1556274968151 = 85557121175 / 2^{39}$.

Це число є значенням е значенням поточного коду. З таблиці 3 визначають відрізок *[0; 6/18)*, якому належить це число і перший закодований символ «s». Віднімаємо від поточного коду нижню границю інтервалу символу «s» і ділимо отриманий результат на ширину цього інтервалу:

$(0,1556274968151-0)/0,3333=0,46692918336$. Це число належить відрізку

[6/18; 10/18], що відповідає символу « і ».

Віднімаємо від поточного коду нижню границю інтервалу символу «і» і ділимо отриманий результат на ширину цього інтервалу:

$(0.46692918336 - 0.3333) / 0.2223 = 0.60112093279$. Це число належить відрізку **[10/18; 13/18]**, що відповідає символу « _ ».

Віднімаємо від поточного коду нижню границю інтервалу символу «_» і ділимо отриманий результат на ширину цього інтервалу:

$(0.60112093279 - 0.5556) / 0.1666 = 0.27323489069$. Це число належить відрізку **[0; 6/18]**, що відповідає символу « s ».

Віднімаємо від поточного коду нижню границю інтервалу символу «s» і ділимо отриманий результат на ширину цього інтервалу:

$(0.27323489069 - 0) / 0.3333 = 0.81978665073$. Це число належить відрізку **[13/18; 15/18]**, що відповідає символу « a ».

Віднімаємо від поточного коду нижню границю інтервалу символу «a» і ділимо отриманий результат на ширину цього інтервалу:

$(0.81978665073 - 0.7222) / 0.1111 = 0.87836769333$. Це число належить відрізку **[15/18; 17/18]**, що відповідає символу « p ».

Віднімаємо від поточного коду нижню границю інтервалу символу «p» і ділимо отриманий результат на ширину цього інтервалу:

$(0.87836769333 - 0.8333) / 0.1111 = 0.40564980495$. Це число належить відрізку **[6/18; 10/18]**, що відповідає символу « і ».

Віднімаємо від поточного коду нижню границю інтервалу символу «і» і ділимо отриманий результат на ширину цього інтервалу:

$(0.40564980495 - 0.3333) / 0.2223 = 0.3254602112$. Це число належить відрізку **[0; 6/18]**, що відповідає символу « s ».

Віднімаємо від поточного коду нижню границю інтервалу символу «s» і ділимо отриманий результат на ширину цього інтервалу:

$(0.3254602112 - 0) / 0.3333 = 0.97647828142$. Це число належить відрізку **[17/18; 1]**, що відповідає символу « , ».

Віднімаємо від поточного коду нижню границю інтервалу символу «_» і ділимо отриманий результат на ширину цього інтервалу:

$(0.97647828142 - 0.9444) / 0.0556 = 0.57694750755$. Це число належить відрізку **[10/18; 13/18]**, що відповідає символу «_».

Віднімаємо від поточного коду нижню границю інтервалу символу «_» і ділимо отриманий результат на ширину цього інтервалу:

$(0.57694750755 - 0.5556) / 0.1666 = 0.12813629982$. Це число належить відрізку **[0; 6/18]**, що відповідає символу «s».

Віднімаємо від поточного коду нижню границю інтервалу символу «s» і ділимо отриманий результат на ширину цього інтервалу:

$(0.12813629982 - 0) / 0.3333 = 0.38444734419$. Це число належить відрізку **[6/18; 10/18]**, що відповідає символу «i».

Віднімаємо від поточного коду нижню границю інтервалу символу «i» і ділимо отриманий результат на ширину цього інтервалу:

$(0.38444734419 - 0.3333) / 0.2223 = 0.23008251997$. Це число належить відрізку **[0; 6/18]**, що відповідає символу «s».

Віднімаємо від поточного коду нижню границю інтервалу символу «s» і ділимо отриманий результат на ширину цього інтервалу:

$(0.23008251997 - 0) / 0.3333 = 0.69031659156$. Це число належить відрізку **[10/18; 13/18]**, що відповідає символу «_».

Віднімаємо від поточного коду нижню границю інтервалу символу «_» і ділимо отриманий результат на ширину цього інтервалу:

$(0.69031659156 - 0.5556) / 0.1666 = 0.80862299855$. Це число належить відрізку **[13/18; 15/18]**, що відповідає символу «a».

Віднімаємо від поточного коду нижню границю інтервалу символу «a» і ділимо отриманий результат на ширину цього інтервалу:

$(0.80862299855 - 0.7222) / 0.1111 = 0.87788477542$. Це число належить відрізку **[15/18; 17/18]**, що відповідає символу «p».

Віднімаємо від поточного коду нижню границю інтервалу символу «р» і ділимо отриманий результат на ширину цього інтервалу:

$(0.87788477542 - 0.8333) / 0.1111 = 0.40130310909$. Це число належить відрізку $[6/18; 10/18]$, що відповідає символу « і ».

Віднімаємо від поточного коду нижню границю інтервалу символу «і» і ділимо отриманий результат на ширину цього інтервалу:

$(0.40130310909 - 0.3333) / 0.2223 = 0.30590692348$. Це число належить відрізку $[0; 6/18]$, що відповідає символу « s ».

Отримаємо такий вираз (Табл. 5).

Таблиця 5 – Декодування повідомлення

| Декодоване число | Символ на виході | Інтервал | Ширина інтервалу |
|------------------|------------------|------------------|------------------|
| 0,1556274968151 | s | $[0; 6/18)$ | 0.3333 |
| 0.46692918336 | i | $[6/18; 10/18)$ | 0.2223 |
| 0.60112093279 | _ | $[10/18; 13/18)$ | 0.1666 |
| 0.27323489069 | s | $[0; 6/18)$ | 0.3333 |
| 0.81978665073 | a | $[13/18; 15/18)$ | 0.1111 |
| 0.87836769333 | p | $[15/18; 17/18)$ | 0.1111 |
| 0.40564980495 | i | $[6/18; 10/18)$ | 0.2223 |
| 0.3254602112 | s | $[0; 6/18)$ | 0.3333 |
| 0.97647828142 | , | $[17/18; 1)$ | 0.0556 |
| 0.57694750755 | _ | $[10/18; 13/18)$ | 0.1666 |
| 0.12813629982 | s | $[0; 6/18)$ | 0.3333 |
| 0.38444734419 | i | $[6/18; 10/18)$ | 0.2223 |
| 0.23008251997 | s | $[0; 6/18)$ | 0.3333 |
| 0.69031659156 | _ | $[10/18; 13/18)$ | 0.1666 |
| 0.80862299855 | a | $[13/18; 15/18)$ | 0.1111 |
| 0.87788477542 | p | $[15/18; 17/18)$ | 0.1111 |
| 0.40130310909 | i | $[6/18; 10/18)$ | 0.2223 |
| 0.30590692348 | s | $[0; 6/18)$ | 0.3333 |

Завдання

- 1. Кодувати окремо вирази 1 і 2 за алгоритмом LZSS, довжинами словника 12, буфера 7.**
- 2. Обчислити довжини закодованих повідомлень і виконати їх порівняння.**
- 3. Визначити коефіцієнти стискання виразів.**
- 4. Виконати декодування повідомлень виразів.**

Вираз 1 : Kein Vorteil ohne Nachteil.

Вираз 2 : Немає переваг без недоліків

- 1. Кодувати окремо вирази 1 і 2 за алгоритмом LZSS, довжинами словника 12, буфера 7.**

Алгоритм LZSS є модифікацією алгоритму LZ77. Робота алгоритму починається однобітовим префіксом, що відділяє код підрядка, що збігається,

від незакодованого символу. Код складається з пари $\langle i, j \rangle$ - зсуву і у словнику підрядка, що збігається з початком буфера, і довжини ѹ цього підрядка. Вікно зсувається на довжину знайденого підрядка або на 1, якщо входження підрядка буфера у словнику не знайдено.

Табл.1 Кодування виразу1
Kein Vorteil ohne Nachteil

| Словник | | | | | | | | | | | | Буфер | | | | | | | Код | l_i | | |
|---------|---|---|---|---|---|---|---|---|---|----|----|-------|---|---|---|---|---|---|--------|--------|------|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | 0'k' | 9 | | |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | 0'e' | 9 | |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | 0'i' | 9 | |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | e | 0'n' | 9 |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | i | 0'_' | 9 |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | t | 0'v' | 9 |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | 1 | 0'o' | 9 |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | 1 | 0'r' | 9 |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | 1 | 0't' | 9 |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | 1<5,2> | 8 | |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | 0'l' | 9 | |
| | | | | | | | | | | | | k | e | i | n | _ | v | o | r | 1<4,1> | 8 | |
| | | | | | | | | | | | | e | i | n | _ | v | o | r | 1<5,1> | 8 | | |
| | | | | | | | | | | | | i | n | _ | v | o | r | t | 0'h' | 9 | | |
| | | | | | | | | | | | | n | _ | v | o | r | t | e | 1<0,1> | 8 | | |
| | | | | | | | | | | | | v | o | r | t | e | i | 1 | 1<5,1> | 8 | | |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|--------|---|
| v | o | r | t | e | i | l | - | o | h | n | e | - | n | a | c | h | t | e | 1<7,1> | 8 |
| o | r | t | e | i | l | - | o | h | n | e | - | n | a | c | h | t | e | i | 1<9,1> | 8 |
| r | t | e | i | l | - | o | h | n | e | - | n | a | c | h | t | e | i | l | 0'a' | 9 |
| t | e | i | l | - | o | h | n | e | - | n | a | c | h | t | e | i | l | 0'c' | 9 | |
| e | i | l | - | o | h | n | e | - | n | a | c | h | t | e | i | l | | 1<5,1> | 8 | |
| i | l | - | o | h | n | e | - | n | a | c | h | t | e | i | l | | | 0't' | 9 | |
| l | - | o | h | n | e | - | n | a | c | h | t | e | i | l | | | | 1<5,1> | 8 | |
| - | o | h | n | e | - | n | a | c | h | t | e | i | l | | | | | 0'i' | 9 | |
| o | h | n | e | - | n | a | c | h | t | e | i | l | | | | | | 0'l' | 9 | |

Табл.2 Кодування виразу 2

Немає переваг без недоліків

| Словник | | | | | | | | | | Буфер | | | | | | | | Код | l_i | |
|---------|---|---|---|---|---|---|---|---|---|-------|----|---|---|---|---|---|---|-----|--------|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'н' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'e' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'm' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'a' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'e' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'п' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 1<6,1> | 8 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'p' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 1<4,1> | 8 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'v' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 1<4,1> | 8 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'г' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 1<4,1> | 8 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'б' | 9 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 1<4,1> | 8 |
| | | | | | | | | | | | | н | е | м | а | € | - | п | 0'з' | 9 |
| | | | | | | | | | | | | н | е | р | е | в | а | г | 1<0,1> | 8 |
| | | | | | | | | | | | | н | е | р | е | в | а | г | 0'h' | 9 |
| | | | | | | | | | | | | н | е | р | е | в | а | г | 1<0,1> | 8 |
| | | | | | | | | | | | | н | е | д | о | л | и | к | 0'd' | 9 |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|------|---|
| е | в | а | г | — | б | е | з | — | н | е | д | о | л | і | к | і | в | | 0'о' | 9 |
| в | а | г | — | б | е | з | — | н | е | д | о | л | і | к | і | в | | 0'л' | 9 | |
| а | г | — | б | е | з | — | н | е | д | о | л | і | к | і | в | | | 0'i' | 9 | |
| г | — | б | е | з | — | н | е | д | о | л | і | к | і | в | | | | 0'к' | 9 | |
| — | б | е | з | — | н | е | д | о | л | і | к | і | в | | | | | 1<6,1> | 8 | |
| б | е | з | — | н | е | д | о | л | і | к | і | в | | | | | | 0'в' | 9 | |

2. Обчислити довжини закодованих повідомлень і виконати їх порівняння.

Довжина ј двійкового коду довжини підрядка, що збігається, є округлений до більшого цілого $\lceil \log_2 F \rceil$, а довжина коду зсуву i – округлений до більшого цілого $\lceil \log_2 (N - F) \rceil$.

довжина $j = \lceil \log_2 F \rceil = \lceil \log_2 7 \rceil = 3$

довжина коду зсуву $i = \lceil \log_2 (N - F) \rceil = \lceil \log_2 12 \rceil = 4$

Довжина отриманого коду

$$Lcode(\text{вираз} 1) = \sum_{i=1}^{25} li = 16*9 + 9*9 = 225$$

Довжина нестиснутого повідомлення $L_{\text{ASCII}}(\text{вираз} 1) = 25*8 = 200$

Довжина отриманого коду

$$li = \textcolor{red}{i}$$

$$Lcode(\text{вираз} 2) = \sum_{i=1}^{27} \textcolor{red}{i} 19*9 + 8*8 = 235$$

Довжина нестиснутого повідомлення $L_{\text{ASCII}}(\text{вираз} 2) = 27*8 = 216$

3. Визначити коефіцієнти стискання виразів.

$$k1(\text{для виразу} 1) = Lcode(\text{вираз} 1) / L_{\text{ASCII}}(\text{вираз} 1) = 225/200 = 1.125$$

$$k2(\text{для виразу} 2) = Lcode(\text{вираз} 2) / L_{\text{ASCII}}(\text{вираз} 2) = 235/216 = 1.088$$

4. Виконати декодування повідомлень виразів.

Таблиця 3 - Декодування виразу 1

| Код | Вихід | Словник | | | | | | | | | | | | |
|------|-------|---------|---|---|---|---|---|---|---|---|---|----|----|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | |
| 0'к' | k | | | | | | | | | | | | k | |
| 0'е' | e | | | | | | | | | | | | e | |
| 0'i' | i | | | | | | | | | | | k | e | i |
| 0'n' | n | | | | | | | | | | k | e | i | n |
| 0'_ | — | | | | | | | | k | e | i | n | — | |
| 0'v' | v | | | | | | | k | e | i | n | — | v | |

| | | | | | | | | | | | | | |
|--------|----|---|---|---|---|---|---|---|---|---|---|---|---|
| 0'o' | o | | | | | | k | e | i | n | - | v | o |
| 0'r' | r | | | | | k | e | i | n | - | v | o | r |
| 0't' | t | | | | k | e | i | n | - | v | o | r | t |
| 1<5,2> | ei | | k | e | i | n | - | v | o | r | t | e | i |
| 0'l' | l | k | e | i | n | - | v | o | r | t | e | i | l |
| 1<4,1> | - | e | i | n | - | v | o | r | t | e | i | l | - |
| 1<5,1> | o | i | n | - | v | o | r | t | e | i | l | - | o |
| 0'h' | h | n | - | v | o | r | t | e | i | l | - | o | h |
| 1<0,1> | n | - | v | o | r | t | e | i | l | - | o | h | n |
| 1<5,1> | e | v | o | r | t | e | i | l | - | o | h | n | e |
| 1<7,1> | - | o | r | t | e | i | l | - | o | h | n | e | - |
| 1<9,1> | n | r | t | e | i | l | - | o | h | n | e | - | n |
| 0'a' | a | t | e | i | l | - | o | h | n | e | - | n | a |
| 0'c' | c | e | i | l | - | o | h | n | e | - | n | a | c |
| 1<5,1> | h | i | l | - | o | h | n | e | - | n | a | c | h |
| 0't' | t | l | — | o | h | n | e | - | n | a | c | h | t |
| 1<5,1> | e | - | o | h | n | e | - | n | a | c | h | t | e |
| 0'i' | i | o | h | n | e | - | n | a | c | h | t | e | i |
| 0'l' | l | h | n | e | - | n | a | c | h | t | e | i | l |

Таблиця 4 - Декодування виразу2

| Код | Вихід | Словник | | | | | | | | | | | |
|--------|-------|---------|---|---|---|---|---|---|---|---|---|----|----|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0'h' | н | | | | | | | | | | | | н |
| 0'e' | е | | | | | | | | | | | | е |
| 0'm' | м | | | | | | | | | | н | е | м |
| 0'a' | а | | | | | | | | | н | е | м | а |
| 0'ε' | ε | | | | | | | | н | е | м | а | ε |
| 0'— | — | | | | | | | н | е | м | а | ε | — |
| 0'п' | п | | | | | | н | е | м | а | ε | — | п |
| 1<6,1> | е | | | | | н | е | м | а | ε | — | п | е |
| 0'р' | р | | | | н | е | м | а | ε | — | п | е | р |
| 1<4,1> | е | | | н | е | м | а | ε | — | п | е | р | е |
| 0'в' | в | | н | е | м | а | ε | — | п | е | р | е | в |
| 1<4,1> | а | н | е | м | а | ε | — | п | е | р | е | в | а |
| 0'г' | г | е | м | а | ε | — | п | е | р | е | в | а | г |

| | | | | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1<4,1> | — | м | а | ε | — | п | е | р | е | в | а | г | — |
| 0'б' | б | а | ε | — | п | е | р | е | в | а | г | — | б |
| 1<4,1> | е | ε | — | п | е | р | е | в | а | г | — | б | е |
| 0'з' | з | — | п | е | р | е | в | а | г | — | б | е | з |
| 1<0,1> | — | п | е | р | е | в | а | г | — | б | е | з | — |
| 0'н' | н | е | р | е | в | а | г | — | б | е | з | — | н |
| 1<0,1> | е | р | е | в | а | г | — | б | е | з | — | н | е |
| 0'д' | д | е | в | а | г | — | б | е | з | — | н | е | д |
| 0'о' | о | в | а | г | — | б | е | з | — | н | е | д | о |
| 0'л' | л | а | г | — | б | е | з | — | н | е | д | о | л |
| 0'i' | i | г | — | б | е | з | — | н | е | д | о | л | i |
| 0'к' | к | — | б | е | з | — | н | е | д | о | л | i | к |
| 1<6,1> | i | б | е | з | — | н | е | д | о | л | i | к | i |
| 0'в' | в | е | з | — | н | е | д | о | л | i | к | i | в |

© А.Є.Коваленко, 2017

Навчальне видання

Коваленко Анатолій Єпіфанович

**ПОБУДОВА КОДІВ НА ОСНОВІ ТИПОВИХ
АЛГОРИТМІВ КОДУВАННЯ ДАНИХ**

Методичні вказівки із самостійної роботи студентів з дисципліни «Теорія
інформації і кодування»

Редактор А.Є.Коваленко
Комп'ютерна верстка А.Є.Коваленко

ІПСА НТУУ «КПІ імені Ігоря Сікорського»
03056 Київ, проспект Перемоги, 37, корп.. 35