**Завдання на контрольну роботу з дисципліни**

 **"Сучасна криптографія "**

**|задавання|****Контрольна робота**

***Завдання 1|задавання|.*** *Перевірка необхідних та достатніх умови однозначного декодування|шикування| для заданої кодової системи.*

Для заданого алфавіту побудувати|спорудженої| кодову систему Хаффмена, доповнити її інверсними кодами цієї ж кодової системи (без повторень).

Для отриманої кодової системи побудувати додаткові сегментні класи та перевірити умови теореми Сардінаса-Петерсона.

**Необхідні і достатні умови однозначного декодування**

Нульовим сегментним класом **Seg0** називається кодова система, записана в довільному порядку (наприклад, за зростанням довжин кодів:

 li <= lj для j>i, l1 = lmin , ln = lmax ).

Доповнення – частина коду, що доповнює префікс до всього коду. Наприклад, для коду 10110 :

префікс 10, доповнення 110, 10 || 110 = 10110.

**Seg1** = **∪** доповнень тих кодів **Seg0** , які мають в якості префікса інші коди **Seg0** .

**Segi** = **∪** доповнень тих кодів **Seg0** , які мають в якості префікса інші коди **Segi-1** , або доповнень тих кодів **Segi-1** , які мають в якості префікса інші коди **Seg0** .

**Теорема** (Сардінас-Петерсон).
 Seg0 є однозначно декодованою кодовою системою ⇔

Seg0 ∩ Segi = ∅ ∀i, 1<= i <= k,
k – кількість додаткових сегментних класів.

***Завдання 2|задавання|.*** *|шикування|Кодування за методом зонного стиску.*

Перекласти задане повідомлення на українську мову і застосувати до одержаного повідомлення метод зонного стиску (крапки, коми не враховувати). Обчислити коефіцієнт стиску.

*Варіанти*.

1. Continue reading this page for highlights of new content in the Library.

2. It contains technical programming information, including sample code.

3. The kit includes build documentation, a hardware reference specification.

4. This article is an introduction to creating your own Microsoft Visio shapes.

5. Drop a Process shape on your drawing and make a modification.

6. Drag the shape onto the blank stencil to create a master.

7. The icon can also be edited to create a more meaningful symbol on the stencil.

8. You can add your shape to the existing stencil by clicking Yes on the above error message.

9. Select the shape and choose Special from the Format menu to enter the copyright information.

10.On the Insert menu, click Picture, and then select a graphic image to insert in your drawing.

11. This turns it into a Visio shape so you can then change its components or add text.

12. Drag the object's selection handles to crop, or drag inside the border to pan the object.

13. Click the arrow next to the Rectangle tool to select the Rectangle or Ellipse tool.

14. The shape will take on an opaque fill if it is successfully closed.

15. The Pencil tool can draw lines and arcs, in addition to the separate Line and Arc tools.

16. The following steps show how to create a repeated series of shapes with equal spacing.

17. To create a repeated series of lines or shapes with equal spacing.

18. When you glue something to a shape's vertex, Visio creates a connection point.

19. The following steps show how to add, move, and delete a new connection.

20. Visio adds a connection point to that location to show the new connection point.

21. A Connection object represents a unique session with a data source.

22. This abstract class defines parameters for asymmetric cryptographic algorithms.

23. Creates a cryptographic object to perform the asymmetric algorithm.

24. Many public-key systems can also be used to form digital signatures.

25. Returns the name of the current Visual FoxPro resource file.

**Метод зонного стиску інформації**

При записі інформації в пам’яті ПК кожній букві або цифрі відводиться зазвичай 8 двійкових розрядів - 1 байт, бо останній є найменшою структурною одиницею пам’яті, що адресується.

Необхідність стиску буквеної інформації напрошується вже виходячи з того, що 8 двійкових розрядів, що складають байт, дозволяють закодувати двійковим кодом алфавіт з 256 букв, тоді як реальні алфавіти, разом з цифрами й допоміжними символами зазвичай не перевищують 50-60 знаків. Однак й 50-60 знаків вимагають  двійкових комбінацій або 5-6-бітову структуру комірки. Така структура не вирішує проблем, оскільки в 2-3-х бітах, що залишились, можна записати лише 4-8 букв (не кажучи вже про проблему зчитування).

Спробуємо тепер використовувати ½ байта для представлення  букв деякого абстрактного алфавіту, потім кодувати інформацію в цьому 16-ти-буквеному (m=16) алфавіті по n=2 букви в кодовому слові. Тоді, використовуючи один байт, можна буде передати ті ж  символів.

Якщо цей 16-ти-буквений алфавіт побудувати таким чином, щоб 13 якісних ознак використовувати як основні символи, а 3 як допоміжні, то можна побудувати наступний алфавіт:

|  |  |
| --- | --- |
| буква | Код |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

Перші 13 символів будемо умовно називати ЦИФРА, а решту 3 – ЗОНА (відношення «цифр» та «зон» обумовлюється кількістю кодових слів у вторинному алфавіті і може змінюватися від 8:8 до 15:1).

Кодові слова у вторинному алфавіті будемо в подальшому будувати таким чином, що перші 4 розряди завжди будуть представляти зону, а інші 4 – цифри. Число можливих комбінацій вторинного алфавіту в цьому випадку зменшиться і буде дорівнювати N=3⋅13=39 (4 зони дали б 12⋅4=48; п’ять – 11⋅5=55 і т.д.), але такий вторинний алфавіт дає принципову можливість побудови 4-бітової структури пам’яті і дозволяє застосувати зонний метод стиску інформації.

Ідея зонного стиску базується на тому, що букви вторинного алфавіту розбиваються по зонах, і якщо в тексті поруч зустрічаються букви, що належать одній зоні, то номер її вказують тільки перед першою буквою, а запис наступних букв обмежується записом їх цифрової частини.

Якщо окрім частоти появи окремих букв в тексті врахувати ще й ймовірність різних буквених сполучень, то кодові слова у вторинному алфавіті будуть мати вигляд, представлений в таблиці 1 (ймовірності вказують тільки частоту букви в тексті).

|  |  |  |  |
| --- | --- | --- | --- |
| зона | D | E | F |
| код | буква | Імовірність | код | буква | Імовірність | код | буква | Імовірність |
| 1 | D0 | пробіл | 0,122 | E0 | У | 0,032 | F0 | Щ | 0,008 |
| 2 | D1 | О | 0,09 | E1 | Д | 0,026 | F1 | Ж | 0,007 |
| 3 | D2 | А | 0,074 | E2 | П | 0,026 | F2 | Ц | 0,006 |
| 4 | D3 | И | 0,059 | E3 | М | 0,023 | F3 | Ю | 0,006 |
| 5 | D4 | І | 0,055 | E4 | Ь | 0,021 | F4 | Ї | 0,006 |
| 6 | D5 | Н | 0,053 | E5 | З | 0,018 | F5 | Є | 0,003 |
| 7 | D6 | В | 0,047 | E6 | Й | 0,017 | F6 | Ф | 0,002 |
| 8 | D7 | Т | 0,044 | E7 | Б | 0,016 | F7 | . |  |
| 9 | D8 | Е | 0,041 | E8 | Я | 0,015 | F8 | ; |  |
| 10 | D9 | Р | 0,040 | E9 | Г | 0,013 | F9 | : |  |
| 11 | DA | С | 0,034 | EA | Ч | 0,012 | FA | ! |  |
| 12 | DB | Л | 0,034 | EB | Ш | 0,010 | FB | ? |  |
| 13 | DC | К | 0,032 | EC | Х | 0,008 | FC | – |  |

Коефіцієнт стиску визначається як відношення кількості байтів в початковому тексті n1 до кількості байтів в стиснутому тексті n2 :

Кст = n1 / n2 .

*Наприклад*, слово “стиску” кодується кодовою послідовністю

DA 7 3 A С E0.

Кст = 6 / 4 = 1,5.

***Завдання 3|задавання|.***

1. Скласти криптографічний протокол для заданої задачі інформаційного обміну.
2. Кожен крок протоколу представити в описовій формі й формально.

 *Варіанти завдань*

1. Передачі даних (ключової інформації або зашифрованої).

2. Шифрування даних на основі симетричного алгоритму.

3. Шифрування даних на основі асиметричного алгоритму.

4. Аутентифікації пароля на основі асиметричного алгоритму.

5. Аутентифікації (перевірки) ключа на основі асиметричного алгоритму.

6. Ідентифікації користувача системи на основі асиметричного алгоритму.

7. Аутентифікації пароля на основі симетричного алгоритму.

8. Аутентифікації (перевірки) ключа на основі симетричного алгоритму.

9. Аутентифікації на основі хеш-функції.

10. Ідентифікації користувача системи на основі хеш-функції.

11. Аутентифікації на основі розділення секрету (знання с нульовим розголошенням).

12. Передачі даних (ЕЦП) на основі симетричного шифрування.

13. Передачі даних (ЕЦП) на основі асиметричного шифрування.

14. Ідентифікації користувача системи на основі ЕЦП.

15. Ідентифікації користувача системи на основі хеш-функції.

16. Передачі даних (хеш-коду).

17. Шифрування даних на основі хеш-функції.

18. Шифрування даних на основі ЕЦП.

19. Аутентифікації пароля на основі хеш-функції.

20. Ідентифікації користувача системи на основі розділення секрету (знання с нульовим розголошенням).