

1. Od czego zależy ilość informacji pojedynczego komunikatu wg teorii Shannona? Jakie są jednostki ilości informacji?

- ilość informacji w komunikacie jest liczbą nieujemną i zależy tylko od prawdopodobieństwa wystąpienia tego komunikatu – im mniejsze prawdopodobieństwo, tym więcej informacji w nim zawartej;

gdy prawdopodobieństwo wystąpienia komunikatu wynosi 1, to ilość informacji w nim zawartej wynosi 0;

- jednostki ilości informacji:
Gdy $a = 10 \rightarrow$ dit
Gdy $a = e$ (liczba Eulera) \rightarrow nit
Gdy $a = 2 \rightarrow$ bit

2. Wyjaśnij pojęcie (czyli podaj wzór i go zinterpretuj) entropii informacyjnej.

$$H(k_1, k_2, k_3, \dots, k_m) = - \sum_{i=1}^m p_i \cdot \log_2 p_i$$

Entropią źródła emitującego m komunikatów k_i z prawdopodobieństwami p_i , $i = 1, 2, \dots, m$ nazywamy średnią (ważoną) ilość informacji zawartą w komunikatach emitowanych przez to źródło.

3. Wyjaśnij pojęcia: słowo kodowe oraz średnia długość słowa kodowego.

- **Słowo kodowe** – zestaw znaków posiadający interpretację.
- **Średnią długością słowa kodowego** – nazywamy średnią ważoną długość komunikatów emitowanych przez to źródło.
Średnia długość słowa kodowego służy do określenia na ile optymalny pod względem efektywności stosowania jest dany kod. Im mniejsza średnia długość słowa kodowego, tym efektywniejszy kod.

4. Czym jest kod jednoznaczny? Jakie jest jego znaczenie w informatyce.

Kod jednoznaczny – taki kod, w którym każdy komunikat można odczytać tylko na jeden sposób.

Kod jest jednoznaczny wtedy, gdy żadne słowo kodowe nie jest początkiem innego.

- precyzja

5. Wyjaśnij pojęcie (czyli podaj wzór i go zinterpretuj) redundancji kodu.

$$R = L - H$$

Redundancja informuje o tym na ile dany kod (o średniej długości słowa kodowego L) można zoptymalizować pod kątem efektywności. Nieprzekraczalną granicą dla L jest wartość H gdyż: jeżeli $L < H$ (czyli $R < 0$), to kod nie może być jednoznaczny. Takie kody nie mają praktycznego znaczenia w informatyce.

6. Jakie znasz inne rodzaje kodów, oprócz jednoznacznego? Wymień i opisz.

- **muzyczny**
- **Alfabet Morse'a** - znaki reprezentowane są przez kilkuelementowe serie sygnałów – krótkich (kropek) i długich (kresek);
- **ASCII** - siedmiobitowy system kodowania znaków, Przyporządkowuje liczbom z zakresu 0–127: litery alfabetu łacińskiego języka angielskiego, cyfry, znaki przestankowe i inne symbole;
- **migowy** - kodowany manualnie wizualno-przestrzenny odpowiednik (subkod) polszczyzny pisanej;

7. Jakie znasz jednostki długości słów kodowych?

- bit

8. Jaką własność ma kod zwarty? Podaj algorytm generowania takiego kodu. Jakie znasz zastosowania takiego kodu?

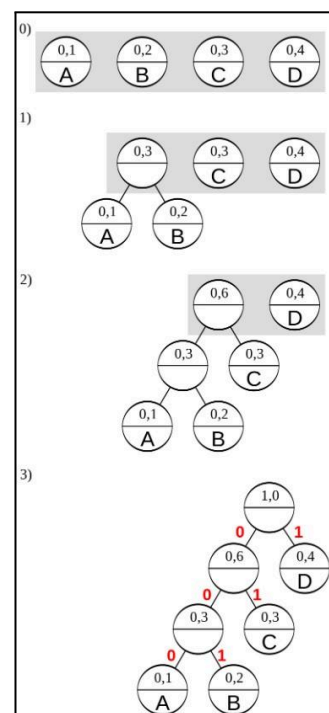
1. **Kod zwarty** to taki kod, w którym żaden komunikat nie jest początkiem innego komunikatu.

2. Algorytm Huffmana

Dany jest zbiór symboli $S=\{A,B,C,D\}$ o prawdopodobieństwach wystąpienia odpowiednio $x=\{0,1;0,2;0,3;0,4\}$.

Kodowanie alfabetu źródłowego zgodnie z algorytmem (oznaczenia jak na rysunku obok):

- Łączenie symboli (A) i (B), co powoduje powstanie $(A + B) = 0,3$; $(C) = 0,3$; $(D) = 0,4$.
- Łączenie drzewa $(A + B)$ z symbolem (C), uzyskując elementy $((A + B) + C) = 0,6$ i $(D) = 0,4$.
- Łączenie drzewa $((A + B) + C)$ z symbolem (D). Teraz pozostaje tylko jeden wolny węzeł (korzeń) – otrzymujemy drzewo $((((A + B) + C) + D) = 1,0$.
- Pozostaje obliczenie kodów poszczególnych symboli:
 - A = lewo, lewo, lewo = 000
 - B = lewo, lewo, prawo = 001
 - C = lewo, prawo = 01



iv. $D = \text{prawo} = 1$

3. **zastosowanie:** kryptoanaliza; kompresja bezstratna

9. Jaką własność ma kod z kontrolą parzystości, a jaką kod samokorygujący się?

- **kod z kontrolą parzystości:** liczba jedynek w każdym słowie kodowym jest parzysta
- **kod samokorygujący (się):** posiada dodatkowe bity pozwalające zidentyfikować pojedynczy bit słowa kodowego, w którym doszło do przekłamania

10. Opisz kod Aikena oraz kod Graya.

- **Kod Aikena** – zamienia parzystą liczbę komunikatów w binarne słowa kodowe, wśród których prawdopodobieństwo wystąpienia wartości 0 na każdym bicie słów kodowych jest równe prawdopodobieństwu wystąpienia wartości 1. Dla dziesięciu komunikatów (np. dla 10 cyfr) są to następujące 4-bitowe słowa kodowe:

Komunikat (wartość dziesiętna)	0	1	2	3	4	5	6	7	8	9
Kod Aikena	0000	0001	0010	0011	0100	1011	1100	1101	1110	1111

- **Kod Graya** – kod, w którym następujące po sobie słowa kodowe (np. odpowiadające kolejnym liczbom) różnią się na jednym bicie; dotyczy to także słów kodowych pierwszej i ostatniej liczby z zakresu.

liczba	Kod Graya		
	1-bitowy	2-bitowy	3-bitowy
0	0	00	000
1	1	01	001
2		11	011
3		10	010
4			110
5			111
6			101
7			100

11. Co oznacza pojęcie liczba kontrolna?

Liczba uzyskana według specjalnego algorytmu służąca do zapewnienia integralności danych.

12. Czym różnią się systemy liczbowe pozycyjne od niepozycyjnych (addytywnych)? Wymień przykłady takich systemów liczbowych.

- **Systemy liczbowe pozycyjne** - sposoby reprezentowania liczb przy użyciu skończonego zestawu cyfr, w taki sposób, że znaczenie cyfry nie jest stałe, lecz zależne od jej pozycji w zapisie liczby.

- binarny (dwójkowy)
- oktalny (ósemkowy)
- decymalny (dziesiętkowy)
- heksadecymalny (szesnastkowy)
- **Systemy liczbowe niepozycyjne (addytywne)** - sposoby reprezentowania liczb przy użyciu skończonego zestawu cyfr, w taki sposób, że znaczenie każdej cyfry jest stałe (niezależne od jej pozycji w zapisie liczby). Słowo „addytywny” oznacza, że wartość danej liczby określa się na podstawie sumy wartości reprezentowanych przez cyfr użyte do zapisu liczby.
 - egipski
 - rzymski
 - grecki

13. Ile istnieje różnych komunikatów n-bitowych?

Na n bitach można zakodować maksymalnie 2^n różnych liczb (komunikatów).

14. Co oznaczają nazwy ASCII, Unicode, UTF, GSM-7?

- **ASCII** - siedmiobitowy system kodowania znaków, Przyporządkowuje liczbom z zakresu 0–127: litery alfabetu łacińskiego języka angielskiego, cyfry, znaki przestankowe i inne symbole
- **Unicode** – standard o rozmiarze 216 x 216 pól przeznaczonych do przechowywania różnych znaków – każdy znak posiada oficjalną nazwę, numer (numer binarny znaku to 16-bitowy numer wiersza + 16-bitowy numer kolumny), a także podstawowy wzorzec graficzny.
- **UTF** – Unicode Transformation Format
- **GSM-7** to standard kodowania znaków używany w komunikacji w sieciach telefonii komórkowej. Jego główną cechą jest ograniczenie się do kodowania 7 bitów na znak.

15. Z jakiego przedziału liczbowego pochodzą liczby całkowite zapisane na n bitach, jeśli wiadomo, że każda liczba ma unikalną reprezentację a ponadto:

- a. są to liczby nieujemne?
- b. są to liczby zarówno ujemne jak i nieujemne i ujemnych jest tyle samo co nieujemnych?

Ad.a) $[0; 2^n-1]$

Ad.b) $[-2^{n-1}; 2^{n-1}-1]$

16. Jak wygląda reprezentacja liczb w formacie stałopozycyjnym, a jak w formacie zmiennopozycyjnym?

- **Stałopozycyjny:** cz. całkowita , cz. niecałkowita
- **Zmiennopozycyjny:** mantysa * podstawa_systemu^{cecha} $\equiv mEc$

17. Dla liczb w formacie zmiennopozycyjnym opisz za pomocą listy kroków
- sposób ich dodawania
 - sposób ich mnożenia

Ad.a)

$$1,2 \cdot 10^3 - 54,11 \cdot 10^2 = 12 \cdot 10^2 - 54,11 \cdot 10^2 = (12 - 54,11) \cdot 10^2 = -42,11 \cdot 10^2$$

Aby dodać / odjąć liczby zmiennopozycyjne należy sprowadzić je do wspólnej cechy a następnie dodać / odjąć mantysy.

Ad.b)

$$1,2 \cdot 10^3 \cdot 54,11 \cdot 10^2 = 1,2 \cdot 54,11 \cdot 10^3 \cdot 10^2 = (1,2 \cdot 54,11) \cdot 10^{3+2} = 64,932 \cdot 10^5$$

Aby pomnożyć / podzielić liczby zmiennopozycyjne należy pomnożyć / podzielić mantysy i dodać / odjąć cechy.

18. Które z praw arytmetyki dokładnej nie zachodzą w arytmetyce maszynowej? Co jest tego przyczyną?

Arytmetyka maszynowa (zarówno liczb stałopozycyjnych jak i zmiennopozycyjnych) różni się od arytmetyki dokładnej z powodu:

- możliwości wystąpienia przepełnienia
- skończonej dokładności z jaką można reprezentować wartości na skończonej liczbie bitów.

19. Co to jest specyfikacja algorytmu?

Algorytm służący rozwiązaniu ustalonego problemu przejmuje od niego dane wejściowe i dane wyjściowe. Sformułowanie danych wejściowych i danych wyjściowych nazywamy **specyfikacją algorytmu**.

20. Jakie znasz sposoby prezentowania (zapisu) algorytmów?

- Opis słowny
- Zapis multimedialny (opis słowny + prezentacja działań, np. nagrany film)
- Lista kroków wyrażona w języku naturalnym lub umownym (np. obrazkowym)
- Pseudokod (pseudojęzyk) – zwykle wymaga wcześniejszego objaśnienia znaczenia użytych symboli i słów kluczowych
- Schemat blokowy
- Język programowania – algorytm zapisany w języku programowania jest programem.

21. Wymień i wyjaśnij cechy algorytmów

- konieczne
- pożądane.

- **a) konieczne**
 - **skończoność** - ciąg czynności prowadzący do rozwiązania ustalonego problemu musi się składać ze skończonej liczby czynności elementarnych;

- **realizowalność** - fizyczna możliwość wykonania czynności elementarnych; w odniesieniu do algorytmów komputerowych cecha ta to określoność algorytmu i rozumiemy przez to fakt, iż zarówno operacje jak i porządek ich wykonania powinny być ściśle określone, nie zostawiając miejsca na dowolną interpretację użytkownika
- **poprawność** - zgodność wyniku działania algorytmu z danymi wyjściowymi zawsze, gdy spełnione są warunki początkowe.
- **b) pożądane**
 - **uniwersalność** - skuteczność algorytmu nie ogranicza się do tylko jednego problemu, lecz do pewnej klasy problemów;
 - **czytelność opisu (zapisu)** - opis algorytmu powinien uwzględniać specyfikę odbiorcy i być dostosowany do tej specyfiki;
 - **efektywność** - algorytm powinien prowadzić od warunków początkowych do warunków końcowych w jak najprostszy sposób.

22. Opisz budowę algorytmów liniowych, z rozgałęzieniami, cyklicznych.

Algorytmy liniowe – algorytmy, w których każda operacja wykonywana jest dokładnie jeden raz dla każdego zestawu wartości danych wejściowych.

Algorytmy z rozgałęzieniami – algorytmy, w których każda operacja wykonywana jest co najwyżej jeden raz dla każdego zestawu wartości danych wejściowych.

Algorytmy cykliczne – algorytmy, w których pewne operacje lub fragmenty obliczeń wykonywane są więcej niż jeden raz dla pewnych zestawów wartości danych wejściowych. Fragment algorytmu wykonywany wielokrotnie nazywamy pętlą.

23. Wyjaśnij pojęcia: analiza algorytmu, symulacja algorytmu.

Symulacja jako przetestowanie jego działania dla różnych wartości danych wejściowych. Symulacja algorytmu ma często na celu sprawdzenie czy dany algorytm posiada wszystkie konieczne i pożądane cechy dla wybranych do symulacji zestawów wartości danych wejściowych.

Analiza algorytmu najczęściej obejmuje jego symulację, ale ma na celu sprawdzenie własności algorytmu bardziej teoretycznie, dla wszystkich możliwych przypadków a nie tylko dla wybranych do symulacji.

24. Czym są algorytmy rekurencyjne?

Funkcją rekurencyjną (rekursją) nazywamy funkcję, dla której obliczenie wartości w niektórych punktach dziedziny wymaga obliczenia wartości tej funkcji w innych punktach dziedziny.

25. Czym jest stos w informatyce? Podaj przykład jego zastosowania.

Stos to struktura danych, której zasada działania polega na tym że ostatni element dodany do stosu jest pierwszym, który zostanie usunięty.

Stos jest często używany do rozwiązywania różnych problemów w informatyce, takich jak śledzenie wywołań funkcji (call stack) w programach, obsługa nawigacji w przeglądarkach internetowych (history stack), czy też w algorytmach związanych z przetwarzaniem języków programowania.

26. Jaki rodzaj funkcji nazywamy boolowskimi?

Funkcje boolowskie to funkcje, które przyjmują argumenty będące elementami zbioru $\{0, 1\}$ lub $\{false, true\}$ i zwracają wartości również z tego zbioru. Innymi słowy, są to funkcje, które operują na zmiennych logicznych i zwracają wynik logiczny.

27. Podaj definicje 16-tu funkcji boolowskich dwóch zmiennych.

Każde wyrażenie boolowskie dwóch zmiennych jest równoważne z którąś z 16 funkcji boolowskich dwóch zmiennych.

28. Podaj przykłady zastosowań funkcji boolowskich w informatyce

- projektowanie bramek logicznych używanych do budowy układów cyfrowych, takich jak procesory
- w systemach operacyjnych do zarządzania uprawnieniami, planowania procesów, zarządzania pamięcią

29. Rozpoznaj postacie normalne funkcji boolowskiej.

$$F: \times_{i=1}^n \{0, 1\} \rightarrow \{0, 1\}.$$

30. Czym jest implikant pierwszy funkcji boolowskiej? Czym pojęcie implikantu pierwszego różni się od pojęcia implikantu?

Implikantem pierwszym funkcji boolowskiej F nazywamy każdy implikant tej funkcji o tej własności, że **jeśli usuniemy z niego dowolny argument, to przestanie on być implikantem**.

Implikantem funkcji boolowskiej F nazywamy taki zestaw (zbiór) argumentów tej funkcji lub ich zaprzeczeń, że **jeśli ich iloczyn ma wartość 1, to wtedy również wartość całej funkcji F ma wartość 1**.

31. Dokonaj przekształcenia funkcji boolowskich do prostszej postaci, np. wyznaczenia implikantów pierwszych.

przykład:

Wartość funkcji $F(a, b, c, d) = 1$ gdy $a \leq d$ i $c \leq d$ lub gdy $b = c$

Zdefiniujmy implikaty pierwsze:

1. Jeśli $a \leq d$ i $c \leq d$, to $F(a, b, c, d) = 1$.
2. Jeśli $b = c$, to $F(a, b, c, d) = 1$.

Teraz możemy połączyć te implikaty w postać bardziej zredukowaną:

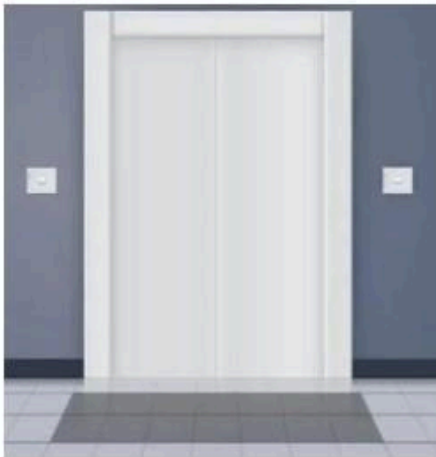
$$F(a, b, c, d) = (a \leq d \wedge c \leq d) + (b = c)$$

To jest już zredukowana forma funkcji, której wartość wynosi 1 dla podanych warunków.

32. Narysuj schemat logiczny odpowiadający zadanej funkcji boolowskiej.

Przykład 2 (zastosowanie funkcji boolowskich w elektronice/automatyce)

Zaprojektuj układ elektroniczny służący otwieraniu drzwi do windy wg następującej specyfikacji: dla wygody użytkowników przyciski otwierające drzwi (P1, P2) umieszczono po obu stronach drzwi, a dla bezpieczeństwa przed otwarciem drzwi przez dzieci w podłodze umieszczono czujnik wagi (P3) - osoby ze zbyt niską wagą nie otworzą drzwi.



Przykład 2 – kontynuacja

Schemat układu cyfrowego realizującego zadanie z przykładu 2



Rozwiązanie: $F(P1, P2, P3) = (P1 \text{ OR } P2) \text{ AND } P3$

$F(P1, P2, P3) = 1$ uruchamia mechanizm otwierania drzwi.

33. Dla podanych liczb wykonaj operacje bitowe.

WDI wykład5.pdf

34. Wyjaśnij różnice pomiędzy procesami współbieżnymi i równoległymi. Co to są procesy współbieżne z przeplotem?

Procesy / wątki współbieżne – procesy / wątki wykonywane w taki sposób, że rozpoczęcie jednego z nich następuje przed zakończeniem drugiego.

- z przeplotem, np. na jednym procesorze: $(X1, Y1, X2, Y2, X3)$ gdzie $X = (X1, X2, X3)$ i $Y = (Y1, Y2)$

Procesy / wątki równoległe – wykonanie procesów / wątków X i Y może odbywać się równocześnie na różnych procesorach, (w szczególności na różnych

komputerach), gdyż procesy / wątki te nie korzystają ze wspólnych zasobów ani żaden z nich nie korzysta z wartości modyfikowanych bądź zwracanych przez drugi.

35. Na czym polega programowanie współbieżne w środowisku rozproszonym?

Sposób programowania umożliwiający współbieżne wykonywanie się procesów na wielu komputerach połączonych w sieć nazywamy **programowaniem współbieżnym w środowisku rozproszonym** lub krócej: **programowaniem rozproszonym**.

36. Wyjaśnij pojęcia: sekcja krytyczna procesu współbieżnego, zasób współdzielony, zmienna współdzielona.

- Fragmenty procesów / wątków korzystające ze zmiennych współdzielonych nazywamy **sekcjami krytycznymi** tych procesów.
- **zasób współdzielony**
 - To obszar pamięci lub inny rodzaj zasobu, który może być dostępny jednocześnie dla wielu procesów lub wątków.
 - "Zadaniem producenta jest wytworzenie produktu, dostarczenie go do magazynu (zasób współdzielony) (...)"
- **zmienna współdzielona** - wspólna zmienna

37. Na czym polega zakleszczenie, a na czym zagłodzenie procesów współbieżnych?

- **Problem zakleszczenia procesów** - sytuacja, w której spośród wielu procesów współbieżnych każdy czeka na zwolnienie wspólnego zasobu przez inny proces, aby móc dokończyć swoje wykonywanie się.
- **Problem zagłodzenia procesu (procesów)** - sytuacja, w której spośród wielu procesów współbieżnych część (co najmniej jeden) oczekuje na przydzielenie zasobów, a te są stale przydzielane pozostałym procesom.

38. Jakie znasz problemy ilustrujące błędy synchronizowania procesów współbieżnych?

1. **Wzajemne Oczekiwanie (Circular Waiting)**: Jest to sytuacja, w której różne procesy czekają na siebie nawzajem w nieskończonym cyklu oczekiwania. Wzajemne oczekiwanie może prowadzić do blokady.
2. **Zakleszczanie zasobów (Resource Locking)**
3. **Głodzenie (Starvation)**