Tablica elementów, w której wyszukujemy połówkowo. Podzielono ją między dwa procesory pracujące z jednakową mocą obliczeniową. Określić przyśpieszenie przy wyszukiwaniu po podziale.

- N elementów w tablicy dzielimy na dwa procesory - każdy dostaje N/2 elementów. Czas potrzebny na przeszukanie (połówkowe) to log(X). Zatem mamy: log(N/2) = log(N) - 1
- Brak przyspieszenia bo $\log_2(N/2)=\log_2(N) - 1$ (teoretycznie algorytm działa szybciej o stałą, ale ma tę samą złożoność)

Jakie działania na liczbach zmiennoprzecinkowych mogą spowodować przepełnienie ?

- dodawanie, mnożenie, dzielenie, odejmowanie, jeśli ich wynik nie może być zapisany przy użyciu dostępnej liczby bitów. (Jak przekraczamy zakres w dół to jest niedomiar, jak w górę to nadmiar).

Podać różnice pomiędzy asemblerem, kodem maszynowym i językiem asemblera ?

- Kod maszynowy – rozkazy numeryczne wykonywane przez procesor
- Asembler – program tłumaczący język asemblera do kodu maszynowego
- Język asemblera – język programowania niskiego poziomu, kody operacji z kodu maszynowego zostały w nim zastąpione mnemonikami

Co to jest: syntaktyka, semantyka ?

Syntaktyka (składnia) – (składnia) - zbiór reguł określający formalnie poprawne konstrukcje językowe [co potrafi zrozumieć kompilator]

Semantyka – opisuje znaczenie konstrukcji językowych, które są poprawne składniowo [do czego służy dana konstrukcja]
Pragmatyka - opisuje, w jaki sposób należy używać tego języka, w jakim celu i przy użyciu jakich reguł praktycznych) [w jakiej sytuacji i w jaki sposób użyć danej konstrukcji]

Problemy nierozwiązywalne i wysoce nierozwiązywalne.

- Łatwo rozwiązywalne: dobra złożoność (wielomianowa), istnieją skuteczne algorytmy
- Trudno rozwiązywalne: złożoność wykładnicza lub gorzej, nie istnieją rozsądne algorytmy
- Problem nierozwiązywalny (nierozstrzygalny): nie istnieje dla niego algorytm który po skończonej liczbie kroków poda jednoznaczna odpowiedz dla dowolnych danych (albo poda błędną odpowiedź albo się zapętli). Przykład: problem stopu, problem domina(dla półpłaszczyzny).
- Problem wysoce nierozwiązywalny (nierozstrzygalny): zadanie sprowadzenia go do problemu nierozstrzygalnego jest samo w sobie nierozstrzygalne

Policzyć ilość informacji w 100-elementowym ciągu liczb z przedziału [1,255], z których każda jest równie prawdopodobna.

100* -(suma od 1 do 255) $1/255*\log_2(1/255) \approx 800$

Wzór Shannona na entropie jednego elementu to zbioru to ($-\sum^{n}_{1} p_i * log_2 p_i$) Czyli suma iloczynu prawdopodobieństwa danego elementu i logarytmu binarnego z tego prawdopodobieństwa dla wszystkich elementów zbioru po kolei, pomnożona przez (-1).

Podać sposób kodowania liczb z zakresu -10 do 1000; ile minimum bitów potrzeba, aby zapisać wszystkie te liczby?

W danym zakresie mamy 1011 liczb (10 ujemnych, 1 zero, 1000 dodatnich)

Przesuwamy zakres o –495. Otrzymamy wtedy zakres symetryczny:
-505..505 => 8=log[2]256<log[2]505<log[2]512=9
Do zapisania takiego zakresu potrzebujemy: 1 bit na zapamiętanie znaku oraz 9 bitów na zapamiętanie wartości
Minimalna liczba bitów niezbędna do zapisania liczb z tego zakresu: 10 bitów

Cechy quicksorta:

- niestabilny (nie zachowuje względnej kolejności)
- złożoność nlogn, pesymistyczna n^2
- wersje iteracyjna i rekurencyjna, wiele wariantów

Jakie to języki imperatywne, aplikatywne i deklaratywne ?

- Imperatywne – pascal, c, c++: program składa się z opisu danych, struktur danych, oraz opisu czynności (funkcji) do wykonania
- Aplikatywne (funkcyjne) – lisp, (haskell, erlang): funkcja jako wartość podstawowa, definiowanie funkcji (często rekurencyjnych), często wzajemnie się wywołują
- Deklaratywne (logiczne) – prolog, sql: opisujemy warunki jakie musi spełniać rozwiązanie, a nie konkretny algorytm jego znalezienia
- Aplikatywne i deklaratywne są wyższego poziomu i mają słabszą efektywność

Podać definicję interpretacji

- Interpretacja – zamiana kodu źródłowego na kod wykonywalny, dokonywana przez interpreter, przy każdym uruchomieniu programu tzn. wykonywanie kodu programu podczas jego ‘czytania’.

Co to znaczy, ze algorytm sortowania tablicy ma złożoność O(n^2) ?

- Liczba operacji (czasu w wypadku złożoności czasowej) potrzebnych do posortowania tablicy przy wzroście jej rozmiaru rośnie tak, jak kwadrat ilości elementów tablicy.

Czym się różni plik sekwencyjny od pliku o swobodnym dostępie?

- Plik sekwencyjny można czytać tylko od początku do końca, dane można dopisywać tylko na koniec takiego pliku, natomiast plik o dostępie swobodnym można czytać i zapisywać w dowolnym miejscu

Czym się różni CPU (procesor) od ALU (jednostki arytmetyczno-logicznej)?

- CPU - urządzenie potrafiące pobierać dane z pamięci, interpretować je i wykonywać jako rozkazy
- ALU - jedna z głównych części procesora prowadząca proste operacje na liczbach całkowitych

Co to jest i do czego służy cache ?

- Pamięć podręczna charakteryzuje się bardzo krótkim czasem dostępu. Jest używana do przechowywania danych, które będą w niedługim czasie przetwarzane. Cache to pamięć statyczna (więcej tranzystorów do 1 bitu informacji)

Algorytmy sortowania

Algorytm	Stabilność	Złożoność	Uwagi

		(nieformalnie)	
bąbelkowe	tak	O(n^2)	
wstawianie	tak	O(n^2)	optymistycznie n
scalanie	tak	O(n $\log_2$ n)	O(n) dodatkowej pamięci
wybieranie	nie	O(n^2)	
combsort	nie	O(n log n)	ulepszony bubble sort
quicksort	nie	O(n log n)	pesymistycznie O(n^2)

Pojęcia: translator, kompilator, debugger, linker, profiler

- Translator to specjalny program komputerowy, dokonujący tłumaczenia programu napisanego w języku programowania, z postaci źródłowej do postaci wynikowej, zrozumiałej dla maszyny.
- Kompilator to program służący do automatycznego tłumaczenia kodu napisanego w jednym języku (języku źródłowym) na równoważny kod w innym języku (języku wynikowym). Proces ten nazywany jest kompilacją.
- Debugger to program komputerowy służący do dynamicznej analizy innych programów, w celu odnalezienia i identyfikacji zawartych w nich błędów.
- Konsolidator (linker) to jeden z programów składowych kompilatora. Konsolidator w trakcie procesu konsolidacji łączy zadane pliki obiektowe i biblioteki statyczne tworząc w ten sposób plik wykonywalny.
- Interpreter to program komputerowy, który analizuje kod źródłowy programu, a przeanalizowane fragmenty wykonuje.
- Profiler to program profilujący lub po prostu profilowanie, to forma dynamicznej analizy programu (przeciwność statycznej analizy kodu). Jest to badanie zachowania programu używając informacji zdobytych podczas jego wykonywania. Zwykle przeprowadza się je by dowiedzieć się, które części programu zoptymalizować, by zwiększyć jego ogólną prędkość lub zmniejszyć wymagania pamięci.

Architektura von Neumanna - rodzaj architektury komputera, przedstawionej przez Johna von Neumanna

- Polega na ścisłym podziale komputera na trzy podstawowe części: procesor, pamięć komputera (wspólna dla danych i programu), urządzenia wejścia/wyjścia

Podejścia do programowania:

- Analityczne - piszemy program używając elementów, które ‘napiszemy później’
- Syntetyczne - piszemy małe kawałki kodu, pomocnicze procedury, a potem sklejamy je w większą całość

Algorytmy, definicje:

- Zadanie algorytmiczne polega na określeniu: wszystkich poprawnych danych wejściowych; oczekiwanych wyników jako funkcji danych wejściowych
- Algorytm - specyfikacja ciągu elementarnych operacji, które przekształcają dane wejściowe na wyniki, algorytm może występować w postaci: werbalnej, symbolicznej (schemat blokowy) programu

Semantyka:

- Semantyka denotacyjna – opis w postaci funkcji przekształcającej dane wejściowe w dane wyjściowe
- Semantyka operacyjna – opis stanu komputera przed i po wykonaniu instrukcji

Instrukcje, typy:

- Proste: przypisania (problemy: zgodność typów, rzutowanie, zakres liczb, nieokreśloność zmiennych, nieobliczalność wyrażenia), wywołania funkcji (problemy: błędna liczba argumentów, niezgodność typu argumentów), operacja we/wy (problemy: konwersja typów)
- Strukturalne: warunkowa, wyboru, iteracyjne

Typy danych:

- skalarny (uporządkowane i skończone zbiory wartości), logiczny, całkowity, rzeczywisty, znakowy, tablicowy

Instrukcje:

- **if** (**<**wyrażenie>) **{** <ciąg instrukcji> **}**
- **if** (<wyrażenie>) **{** <ciąg instrukcji 1> **}**
  else { <ciąg instrukcji 2> **}**
- **switch** (<wyrażenie>) **{**
  case <etykieta1> : <instr1>; break;
  case <etykieta2> : <instr2>; break;
  ...
  default <instr>;
  }
- **while** (<wyrażenie>) { <ciąg instrukcji> }
  do { <ciąg instrukcji> } **while** (<wyrażenie>);
- **while** (True) **{**
  ...
  if (<wyrażenie>) break;
  ...
  }
- **for** (<inst1> ; <war> ; <inst2>) {
  <ciąg instrukcji>;
  }

EBNF:

Użycie	Zapis
definicja	=
złączenie	,
zakończenie	;
alternatywa	|
zawartość opcjonalna	[...]
powtórzenie	{ ... }
grupowanie	(...)
tekst dosłowny	" ... "
tekst dosłowny	' ... '
komentarz	(* ... *)
wyrażenie specjalne	? ... ?
wyjątek	-
wielokrotność	*

Klasa string - metody:

- empty() - zwraca wartość true jeśli napis jest pusty
- size(), length() - zwraca ilość znaków w napisie
- at() - Zwraca znak o podanym położeniu, tak jak operator [], wyjątek w przypadku wyjścia poza zakres
- clear() - usuwa wszystkie znaki z napisu
- erase() - usuwa wybrane znaki

- find() - znajduje podciąg w ciągu
- swap() - zamienia miejscami dwa stringi
- substr() - zwraca podciąg na podstawie indeksu początkowego i długości ciągu
- append() - dodaje zadany napis na końcu istniejącego ciągu
- c_str() - zwraca napis w stylu języka C, (stały wskaźnik typu const char*)

Cele stosowania procedur i funkcji:

- dekompozycja problemu
- wielokrotne wykonanie
- poziomy abstrakcji
- oddzielna kompilacja
- możliwość użycia rekurencji

Maksymy i rady programistyczne:

- Programy mają być czytane przez ludzi
- Czytelność jest zwykle ważniejsza niż sprawność
- Najpierw projekt potem kodowanie
- Dawaj więcej komentarzy niż będzie ci, jak sądzisz potrzeba
- Stosuj komentarze wstępne
- Stosuj przewodniki w długich programach
- Komentarz ma dawać coś więcej, niż tylko parafrazę tekstu programu
- Błędne komentarze są gorsze niż zupełny brak komentarzy
- Stosuj odstępy dla poprawienia czytelności
- Używaj dobrych nazw mnemonicznych
- Wystarczy jedna instrukcja w wierszu.
- Porządkuj listy według alfabetu
- Nawiasy kosztują mniej niż błędy
- Stosuj wcięcia dla uwidocznienia struktury programu i danych

Sortowanie:

- Sortowaniem nazywamy proces ustawiania zbioru obiektów w określonym porządku.
- Metodę sortowania nazywamy stabilną, jeśli podczas procesu sortowania pozostaje **nie zmieniony** względny porządek obiektów o identycznych kluczach
- Klasyfikacja według rodzaju struktury: sortowanie listy, tablicy, łańcucha, pliku
- Klasyfikacja według miejsca sortowania: wewnętrzne, zewnętrzne
- metody intensywne, ekstensywne (wymagają dodatkowej pamięci)

Struktury liniowe o zmiennym podłożu:

- Są to struktury nie posiadające adresacji
- Dostęp do poszczególnych elementów struktury jest organizowany poprzez wyróżnienia.
- Do tych struktur należą: stos, kolejka, talia, lista jednokierunkowa i dwukierunkowa
- Implementacja struktur: tablicowa (szybkość, prosta implementacja, ograniczenia pamięciowe), wskaźnikowa, mieszana

Stos:

- Wyróżnienia: wierzchołek stosu.
- Operacje proste (4 operacje): inicjalizacja stosu – init(s), testowanie czy pusty – empty(s), dołączanie elementu na wierzchołek – push(s, e), pobieranie elementu z wierzchołka – pop(s)
- Uwagi: struktura pracuje jednym końcem, określana jest jako struktura LIFO (Last In First Out)
- Zastosowanie: przeglądanie grafu, obliczania wartości wyrażenia, usuwanie rekurencji z programu

Kolejka:

- Wyróżnienia: początek kolejki, koniec kolejki
- Operacje proste (4 operacje): inicjalizacja kolejki – init(k), testowanie czy pusty – empty(k), dołączanie elementu na koniec – put(k, e), pobieranie elementu z początku – get(k).
- Uwagi: struktura pracuje oboma końcami, określana jest jako struktura FIFO (First In First Out).
- Zastosowanie: przeglądanie grafu, kolejka zadań o jednakowym priorytecie

Postać wyrażenia:

- Postać wyrostkowa, infiksowa: (x+y) - (z*3)
- Postać przedrostkowa, prefiksowa, notacja Łukasiewicza: - (+(x,y), *(z,3))
- Postać przyrostkowa, postfiksowa : x y + z 3 * -

Arytmetyka liczb w maszynie

- różna od arytmetyki używanej przez ludzi - system dwójkowy - skończona i ustalona precyzja
- własności liczb o skończonej precyzji (zakresie) są inne
- zbiór liczb o skończonej precyzji (zakresie) nie jest zamknięty na żadne działanie
- nie działa prawo łączności a+(b+c) <> (a+b)+c
- nie działa prawo rozdzielności mnożenia względem dodawania a*(b+c) <> a*b + a*c

Liczby zmiennoprzecinkowe

- System binarny (typ 4 bajtowy Single)
- mantysa - 23 bity + 1bit na znak, cecha - 8 bitów
- dodatnia wartość maksymalna: 0 111..1*$2^{01111111}$
- ujemna wartość minimalna: 1 111...1*$2^{0111111}$
- dodatnia wartość minimalna: 0 000...1*$2^{10000000}$
- ujemna wartość maksymalna: 1 000...1*$2^{10000000}$

System operacyjny:

- System operacyjny jest to zbiór procedur (programów) przekształcający maszynę rzeczywistą w wirtualną. System operacyjny jest to zorganizowany zespół programów, które pełnią rolę pośredniczącą między sprzętem, a użytkownikami, dostarczają użytkownikom zestawu środków ułatwiających projektowanie, kodowanie, uruchamianie i eksploatację programów oraz w tym samym czasie sterują przydziałem zasobów dla zapewnienia efektywnego działania

Zasoby:

- Zasób systemu: sprzęt lub program, który może być przydzielany systemowi operacyjnemu lub programowi użytkowemu.
- Zasoby programowe: funkcje systemowe dostarczane programowi użytkownika, określone obszary pamięci - bufory, pamięć zewnętrzna, katalogi i pliki, translatory, kompilatory

Tryby pracy systemów operacyjnych:

- systemy do przetwarzania wsadowego (off-line, batch)
- systemy z podziałem czasu (on-line)
- system dla działania w czasie rzeczywistym (real-time

Złożoność obliczeniowa algorytmu:

- ilość zasobów komputerowych potrzebnych do jego wykonania (czas procesora, wielkość pamięci).
- złożoność pesymistyczna (ilość zasobów potrzebnych przy „najgorszych” danych wejściowych o rozmiarze n)
- złożoność oczekiwana (ilość zasobów potrzebnych przy „typowych” danych wejściowych o rozmiarze n),

Funkcja złożoności obliczeniowej:

- Funkcja złożoności obliczeniowej algorytmu rozwiązującego dany problem to funkcja przyporządkowująca każdej wartości rozmiaru konkretnego problemu maksymalną liczbę kroków elementarnych (lub jednostek czasu) komputera potrzebnych do rozwiązania za pomocą tego algorytmu konkretnego problemu o tym rozmiarze.

Złożoność czasowa algorytmu:

- Algorytm wielomianowy (o złożoności czasowej wielomianowej) to taki, którego złożoność jest O(p(n)), gdzie p(n) jest wielomianem, a n jest rozmiarem problemu.
- Algorytm wykładniczy (o złożoności czasowej wykładniczej) to taki, który nie jest wielomianowy.

Problemy algorytmiczne:

- problem najkrótszej drogi, problem komiwojażera, problem tautologii, funkcja Ackermana-Hermesa, problem stopu, problem domina, problem domina-wąż

Fazy kompilacji:

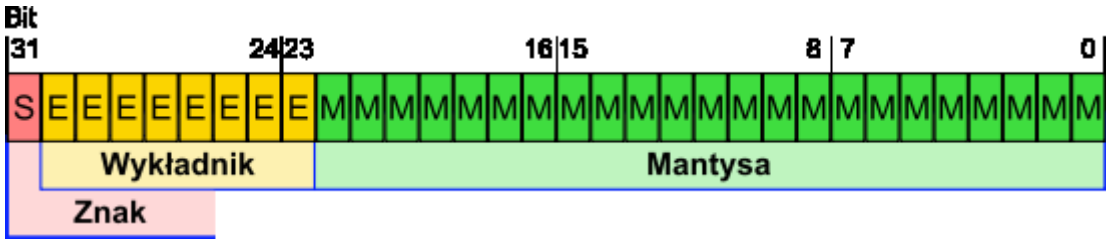
- preprocessing (preprocesor); *Preprocesor – program interpretujący, którego zadaniem jest przetworzenie tekstu wejściowego w sposób określony za pomocą poleceń preprocesora przez programistę na tekst wyjściowy. Dopiero tak przetworzony tekst poddawany jest analizie składniowej i kompilacji.*
- analiza leksykalna (skaner) - wyodrębnia podstawowe jednostki (liczby, słowa); *Lekser (ang. lexer lub scanner), nazywany też analizatorem leksykalnym, to program komputerowy który dokonuje analizy leksykalnej danych wejściowych, zwykle jako pierwsza część jakiegoś większego procesu*
- analiza składniowa (parser) - bada zgodność z gramatyką języka; *Parser (inaczej analizator składniowy) w informatyce program dokonujący analizy danych wejściowych w celu określenia ich gramatycznej struktury w związku z formalną gramatyką.*
- optymalizacja niezależna od architektury

- optymalizacja zależna od architektury
- dołączanie bibliotek (linker)
- generacja kodu

Maszyna wielopoziomowa:

- generator aplikacji
- język wyższego poziomu
- język wysokiego poziomu
- język assemblera
- system operacyjny
- kod maszynowy
- mikroprogram (występował w starszych komputerach teraz to chyba są drivery)
- sprzęt (ostatnie 2 to maszyna rzeczywista)

Kiedy występuje i jak rozpoznajemy nadmiar zmiennoprzecinkowy (cecha i mantysa kodowane są w kodzie U2).



- Nadmiar zmiennoprzecinkowy (overflow) występuje gdy wartość liczby przekroczy swoją maksymalną wartość. Maksymalna liczba jaką można reprezentować: S = 0, E = [1...1], M= [1...1]
- Gdy do takiej liczby dodamy cokolwiek nastąpi przepełnienie i w rezultacie dopełniania do dwóch zmianie na 1 ulegnie bit znaku. Tak więc overflow rozpoznajemy po zmianie znaku liczby (liczba dodatnia nagle staje się ujemna).

Ile informacji zawiera 10 znakowe słowo którego każdy znak z jednakowym prawdopodobieństwem jest jedną z liter a, b, c, d, e, f, g.

$$X = \{a, b, c, d, e, f, g\}$$

$$P(\{a\}) = P(\{b\}) = ... = P(\{g\}) = \frac{1}{7}$$

Zgodnie ze wzorem Claude E. Shannona na entropie informacji zbioru X:

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

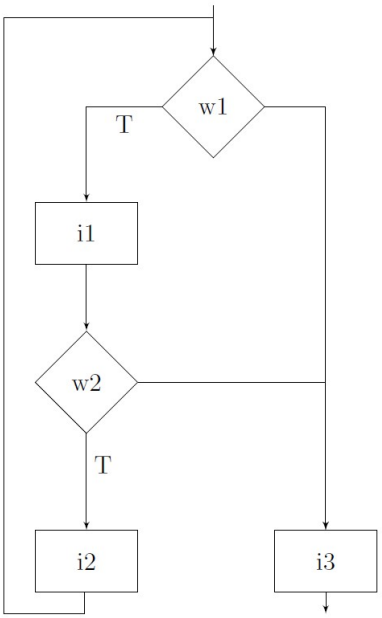
p_i - prawdopodobieństwo wystąpienia i-tego znaku w danym słowie

Nasze słowo składa się z 10 znaków stąd:

$$H(X) = -\sum_{i=1}^{7} \frac{1}{7} \log_2 \frac{1}{7} = -\frac{7}{7} \log_2 \frac{1}{7} = \log_2 7$$ (ilość informacji zawarta w jednym znaku)

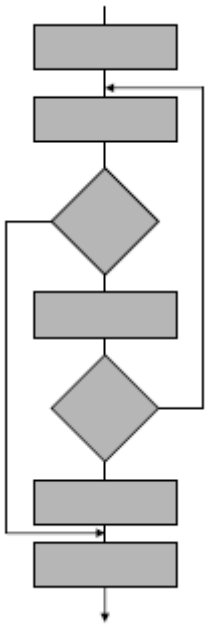
$10H(X) \quad 10 \ \log_2 7 \quad 28,07$

Używając wyłącznie konstrukcji strukturalnych, napisać program odpowiadający poniższemu schematowi.




```
while ( w1 && w2 )
{
i1;
i2;
}
if ( w1 ) { i1 };      //opcjonalnie ( w1 && !w2 )
i3;
```

Zapisać w postaci kodu bez goto



```
zmienna1=false;
instrukcja1;
do {
instrukcja2;
if(!warunek1) zmienna1=true;
if(!zmienna1) instrukcja3;
}
while(!zmienna1 and !warunek2);
if(!zmienna1) instrukcja4;
instrukcja5;
```

Ile wymiarową tablicę można zaimplementować

- Limitem jest wielkość stosu

Co zwraca funkcja abs() dla najbardziej ujemnej liczby?

- Największą możliwą ujemną liczbę w danym typie

Co zwraca operacja mod gdy jeden/oba argumenty są ujemne?

- Dla dzielnej ujemnej wynik jest taki jak (-1)*(reszta z dzielenia wartości bezwzględnej z tej liczby) czyli np. dla -12 mod 5 jest -2, bo 12 mod 5 = 2. Znak minus przy dzielniku nie wpływa na wynik

Które z 6 instrukcji: if , if else, case, while, do while, for można usunąć z języka aby nadal można było w nim programować?

- wystarczy zostawić samego while'a

Do czego służy DMA, wymienić bloku funkcjonalne prostego komputera ?

- technika, w której sprzęt komputerowy podłączony do płyty głównej, mogą korzystać z pamięci operacyjnej RAM lub portów we-wy, pomijając przy tym CPU
- pamięć, CPU, urządzenia we/wy

Masz 21 bitów na cechę i 30 na mantysę; napisać: liczbę największą, najmniejszą, najmniejszą dodatnią

- Jeśli mamy 21 bitów na cechę i zapisujemy ją w U2 to cecha jest w przedziału $[-2^{20} \dots 2^{20}-1]$.
- Zakładam, że w mantysie przeznaczamy jeden bit na znak liczby a 29 bitów na wartość mantysy. Wtedy minimalna znormalizowana (w notacji Icona) mantysa wynosi ½ . Maksymalna natomiast to suma szeregu ½ + ¼ + … + 1 / 2^n w tym wypadku aż do 1/ 2^29 = (1/2) * (1- (1/2)^29)/(1 - 1/2) = 1 - $½^{29}$ co można zaokrąglić do 1.
- W takim razie maksymalna liczba dodatnia, którą możemy zapisać przy znormalizowanej mantysie to: maksymalna mantysa znormalizowana * 2^(maksymalna cecha) - (1 - 0.5^{29}) * 2^(2^{20} -1)
- Minimalna liczba dodatnia, którą możemy zapisać to: minimalna mantysa znormalizowana * 2^(minimalna cecha) - ½ * 2^(-2^20) = 2^(-1) * 2^(-2^20) = $2^{(- 2^{20} - 1)}$

Ile bitów potrzeba do zapisania liczby zmiennoprzecinkowej z przedziału od 10^{-6} do 10^6 z dokładnością do trzech cyfr dziesiętnych?

- Do zapisu w systemie dziesiątkowym mantysy potrzebowalibyśmy 3 cyfr. Mamy więc zależność:
- $2^{-x} = 10^{-3}$ -> $2^x = 10^3$ -> log2(2^x) = log2(10^3) -> x = log2(10^3) -> x = 3 log2(10) -> x = 3/log10(2) ~ 3/0.3010 -> x ~ 10
- Mając mantysę Iconową dodajemy 1 bit na znak liczby, więc potrzeba 11 bitów. Maksymalna mantysa wynosi (1 - ½ ^10) co można przybliżyć jako 1.
- Teraz liczymy cechę, cecha to x, więc 1 * $2^x = 10^6$ -> x ~ 20
- W takim razie musimy przechować liczby [-20, 20] jako cechę
- Biorąc system U2 potrzeba nam 6 bitów na cechę, (to daje zakres [-32 .. 31])
- Razem mamy 6+11 = 17 bitów.

- Znana jest także wersja zadania z dokładnością do 6 cyfr, wtedy wychodzi ~27 bitów