



Algorytm Huffmana

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Animacja](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Algorytm Huffmana

Źródło: Michael Dzedzic, domena publiczna.

Zdarza się, że pewne informacje są zapisywane w postaci nadmiarowej, czyli z wykorzystaniem zbyt dużej liczby bitów. Aby zmniejszyć zużycie zasobów komputera możemy dokonać kompresji danych – to znaczy zakodować informacje w taki sposób, aby zaoszczędzić miejsce w pamięci komputera albo na dysku twardym.

Kompresję danych możemy podzielić na stratną i bezstratną. Jednym z algorytmów kompresji bezstratnej jest algorytm Huffmana. Stosując go, uzyskujemy – po rozkodowaniu danych – informację niezmienną, identyczną jak początkowa. Algorytmu Huffmana używa się jako ostatniego etapu w różnych systemach kompresji, zarówno bezstratnej, jak i stratnej np. w formatach kompresji obrazu i dźwięku, np. JPEG, MPEG oraz MP3.

Implementacje omawianego zagadnienia w poszczególnych językach programowania znajdziesz w e-materiałach:

- [Algorytm Huffmana w języku C++](#),
- [Algorytm Huffmana w języku Java](#),
- [Algorytm Huffmana w języku Python](#).

Więcej zadań? Sięgnij do [Algorytm Huffmana – zadania maturalne](#).

Twoje cele

- Dowiesz się, na czym polega algorytm Huffmana i poznasz listę czynności wykonywanych podczas jego realizacji.
- Przeanalizujesz kodowanie przykładowego wyrazu zgodnie z algorytmem Huffmana.
- Nauczysz się odkodowywać ciąg znaków, korzystając z drzewa Huffmana.

Przeczytaj

Jednym z algorytmów wykorzystywanych w [kompresji bezstratnej](#) jest algorytm Huffmana. Jego nazwa pochodzi od nazwiska pomysłodawcy tej metody, informatyka Dawida Huffmana.

Huffman zwrócił uwagę na fakt, że różne znaki – na przykład litery – możemy zapisywać za pomocą ciągów bitów o długości zależnej od częstości ich występowania. Znaki, które występują najczęściej mają kody najkrótsze, dzięki czemu łączna liczba bitów jest mniejsza niż przy stosowaniu kodów o stałej długości. Pozwala to zmniejszyć zapotrzebowanie na miejsce niezbędne do przechowywania danych.

Algorytm Huffmana – lista kroków

Ideą algorytmu Huffmana jest sprawdzenie liczby wystąpień każdego elementu kompresowanego zbioru. Element pojawiający się najczęściej jest kodowany z użyciem najkrótszego ciągu bitów. Symbolowi występującemu najrzadziej należy przyporządkować najdłuższy ciąg bitów.

Oto lista czynności składających się na algorytm Huffmana:

1. Utwórz listę [drzew binarnych](#), z których każde reprezentuje pojedynczy znak alfabetu (unikatowy element kompresowanego zbioru). Drzewo ma składać się z korzenia zawierającego nazwę znaku oraz liczbę jego wystąpień.
2. Znajdź dwa drzewa z najmniejszą liczbą wystąpień znaków i połącz je w jedno. W korzeniu tego drzewa zapisz sumę liczb z drzew składowych; one same mają być prawym i lewym poddrzewem nowego drzewa. Krawędzi łączącej korzeń z lewym poddrzewem przypisz wartość 0, natomiast krawędzi łączącej korzeń z prawym poddrzewem – wartość 1.
3. Jeżeli na liście jest więcej niż jedno drzewo, przejdź do kroku 2.
4. Utwórz kody poszczególnych znaków alfabetu, przechodząc od korzenia do znaku i łącząc przy tym wartości odpowiadające każdej krawędzi.

Algorytm Huffmana – przykład

Porównajmy kodowanie Huffmana z systemem zapisywania symboli zgodnie ze standardem ASCII, w przypadku którego każdemu znakowi odpowiada unikatowy ciąg złożony z siedmiu bitów. Pozwala to zapisać $2^7 = 128$ różnych symboli.

Zapiszmy słowo ANIA za pomocą kodów ASCII. Musimy zakodować trzy różne litery:

A – dwukrotnie, za pomocą ciągu bitów 01000001

N – za pomocą ciągu bitów 01001110

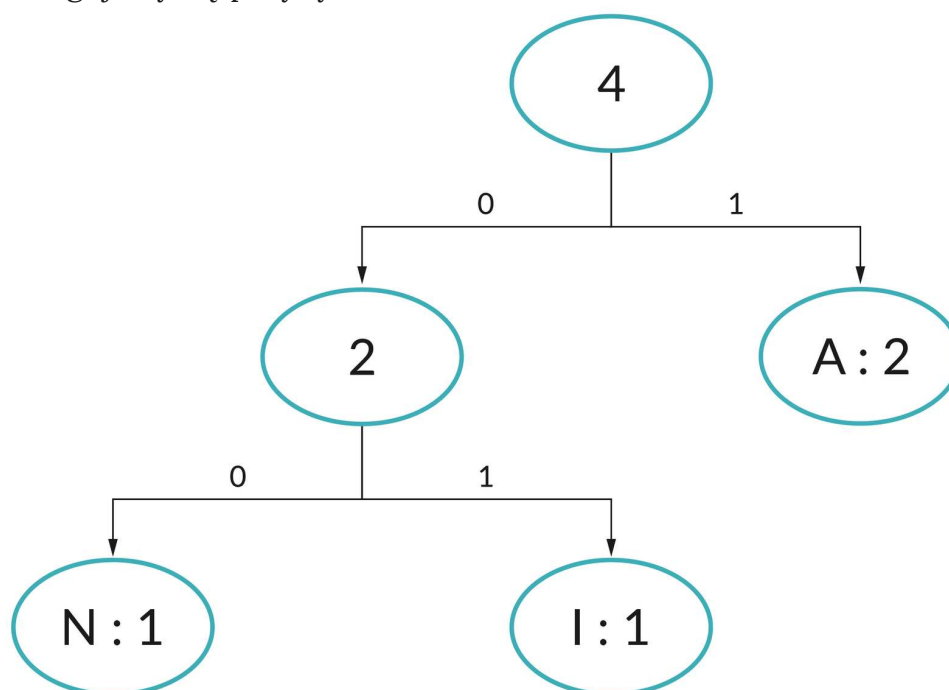
I – za pomocą ciągu bitów 01001001

Po zakodowaniu słowo ANIA ma postać:

01000001 01001110 01001001 01000001

Do zapisania czterech liter użyliśmy 28 bitów.

Algorytm Huffmana pozwala zmniejszyć liczbę bitów wykorzystywanych do zapisania informacji dzięki przyporządkowaniu odpowiednich kodów literom składającym się na słowo ANIA. Posługujemy się przy tym tzw. drzewem Huffmana:



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Korzystając z drzewa odczytujemy kody poszczególnych znaków:

A – 1

N – 00

I – 01

Słowo Ania zakodowane z użyciem algorytmu Huffmana przybiera zatem postać:

100011

W tym przypadku do zapisania wyrazu użyliśmy tylko 6 bitów.

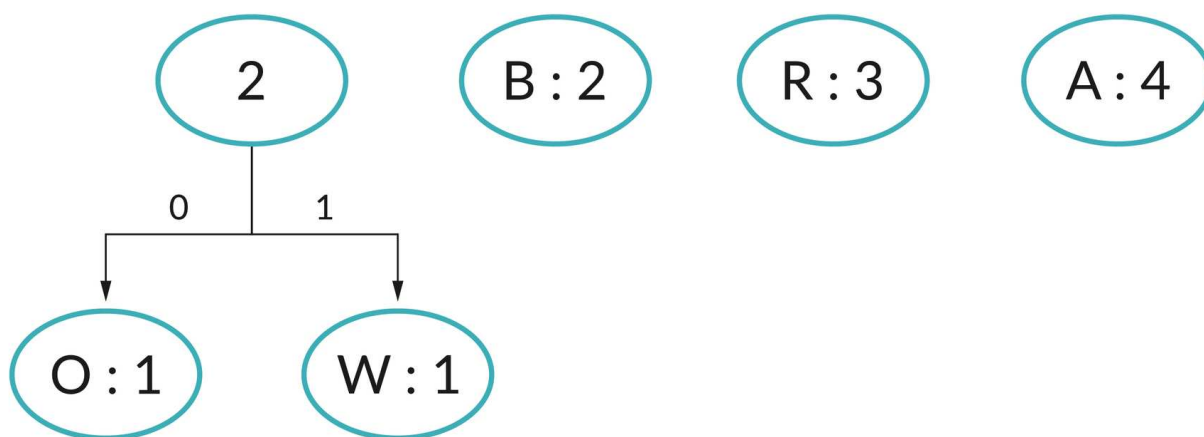
Przykład kodowania Huffmana

Przeanalizujemy teraz przykład kodowania zbioru symboli z wykorzystaniem algorytmu Huffmana. Zakodujemy słowo RABARBAROWA. Każdemu symbolowi przypiszmy liczbę jego wystąpień, a następnie utwórzmy listę drzew binarnych. Każde drzewo zawiera symbol litery i liczbę jej wystąpień.



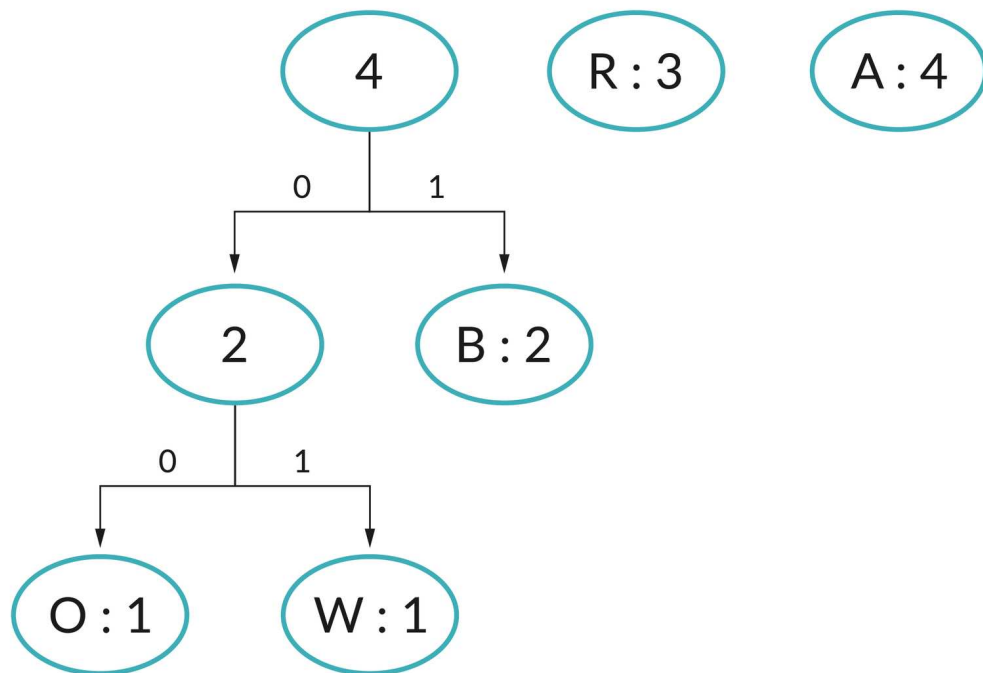
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Dwa drzewa reprezentujące litery występujące najrzadziej – O oraz W – łączymy w jedno drzewo. W korzeniu umieszczamy sumę wystąpień tych symboli, czyli 2. Lewym poddrzewem będzie litera O. Prowadzącej do niego krawędzi przypisujemy wartość 0; prawym poddrzewem jest W, a prowadząca do niego krawędź ma wartość 1:



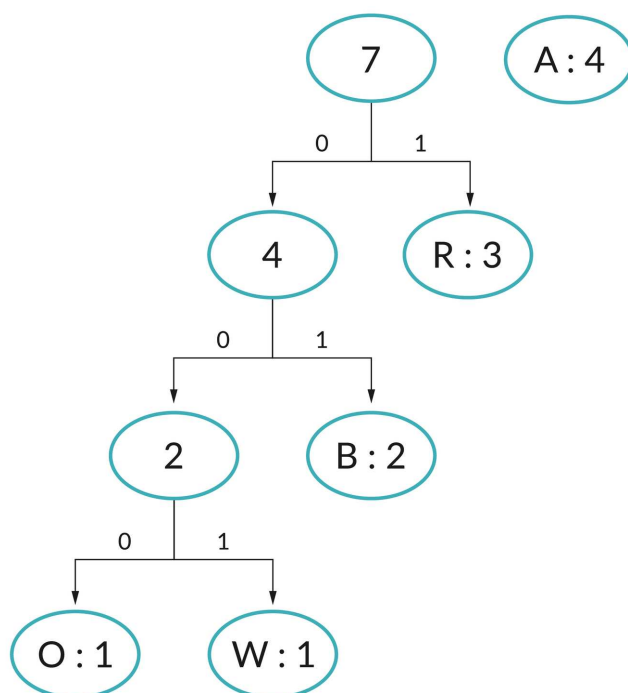
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Pozostały 4 listy. Ponownie łączymy drzewa z zapisaną najmniejszą liczbą wystąpień. W korzeniu umieszczamy sumę wystąpień liter O, W oraz B. Przypisujemy odpowiednie wartości nowym krawędziom drzewa: lewej – 0, natomiast prawej – 1:



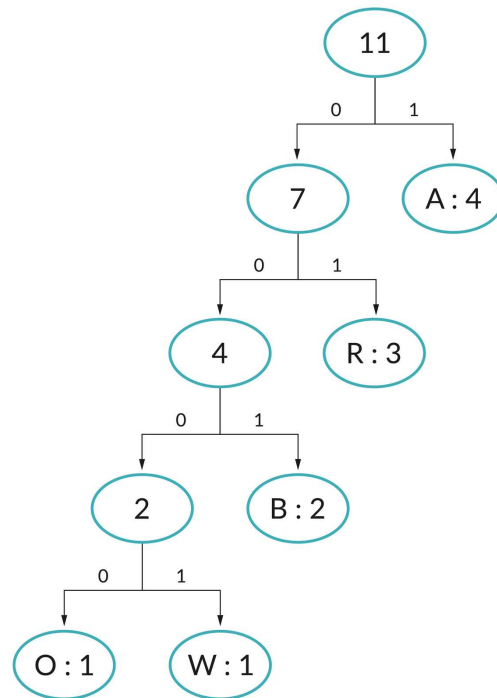
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Na liście nadal jest więcej niż jedno drzewo. Łączymy więc dwa drzewa z najmniejszą liczbą wystąpień – będzie to drzewo składające się z samego korzenia R oraz drzewo z symbolami O, W, B. Sumą wystąpień jest suma znajdująca się w korzeniach dwóch drzew. Pamiętamy o przypisaniu krawędziom wartości 0 lub 1.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Teraz łączymy dwa pozostałe drzewa. W korzeniu umieszczamy sumę wystąpień liter zapisaną w korzeniach, czyli wartość 11. Jest to liczba wszystkich liter, z których składa się kodowane słowo.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Na liście pozostało jedno drzewo. Przystępujemy zatem do odczytania kodów symboli. Przechodzimy w tym celu od korzenia do każdej litery.

Otrzymujemy następujące kody:

R – 01

A – 1

B – 001

O – 0000

W – 0001

Zapisujemy kody w kolejności odpowiadającej literom składającym się na słowo RABARBAROWA. Oto rezultat:

0110011010011010000000011

Do zapisania słowa złożonego z 11 liter użyliśmy zatem tylko 24 bitów (zamiast 77, których musielibyśmy użyć posługując się kodami ASCII).

Kod Huffmana jest **kodem prefiksowym**. Oznacza to, że ciąg bitów opisujący dowolny symbol nie jest początkiem kodu żadnego innego symbolu.

W zaprezentowanym przykładzie kodem litery A jest 1. Nietrudno zauważyć, że żaden inny kod nie zaczyna się od tej cyfry. To samo dotyczy kodu litery R: od ciągu 01 nie rozpoczyna się ani jedna spośród pozostałych sekwencji bitów.

Słownik

kompresja bezstratna

metoda kompresji, w przypadku której – po zdekompresowaniu danych – otrzymujemy informację niezmienną, taką samą jak początkowa (przed skompresowaniem)

drzewo binarne

drzewo, w którym każdy węzeł ma nie więcej niż dwóch synów

kod prefiksowy

kod, w którym żadne ze słów kodowych nie jest początkiem innego słowa

Animacja

Polecenie 1

Zapoznaj się z animacją.



Film dostępny pod adresem </preview/resource/RfayLM2QOGpQk>

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film nawiązujący do treści materiału algorytm Huffmana.

Ćwiczenie 1

Wyjaśnij, dlaczego nie dałoby się odkodować ciągu, gdyby kod dowolnego symbolu był początkiem kodu innego znaku.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Uzupełnij poniższy tekst.

Algorytm Huffmana jest metodą , która polega na liczby zapisanej informacji, .

bezstratnej

delikatnie ją zmieniając

zwiększeniu

symboli

bajtów

kompresji

bez zmiany wersji oryginalnej

słów

stratnej

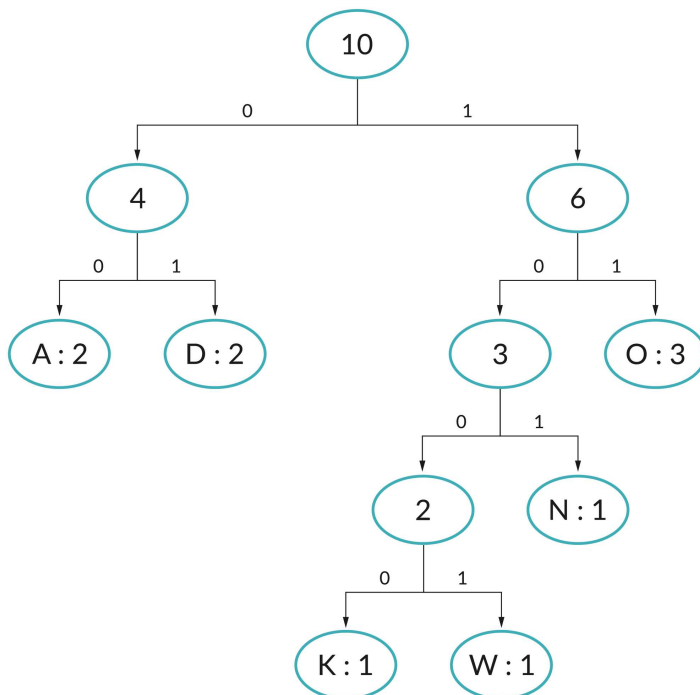
zmniejszeniu

bitów

Ćwiczenie 2



Wskaż, jakie hasło zostało zakodowane za pomocą ciągu bitów 10001101 i z wykorzystaniem poniższego drzewa Huffmana.



☐ KON

☐ WON

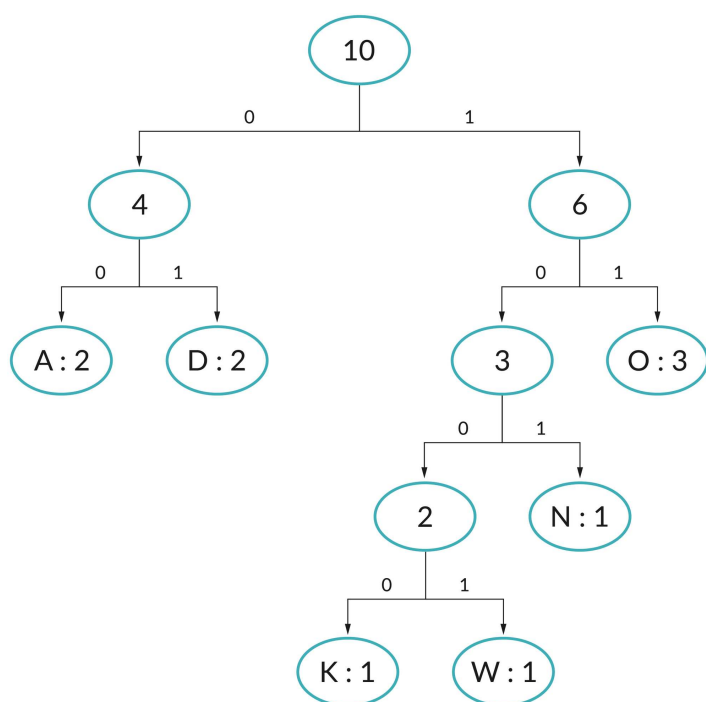
☐ KOD

☐ KWO

Ćwiczenie 3



Zaznacz kod symbolu „W” w poniższym drzewie Huffmana.



☐ 1100

☐ 1011

☐ 1000

☐ 1001

☐ 01

☐ 1010

☐ Drzewo o korzeniu z symbolem O

☐ Drzewo o korzeniu z symbolem Y

☐ Drzewo o korzeniu z symbolem B

☐ Drzewo o korzeniu z symbolem A

4

2

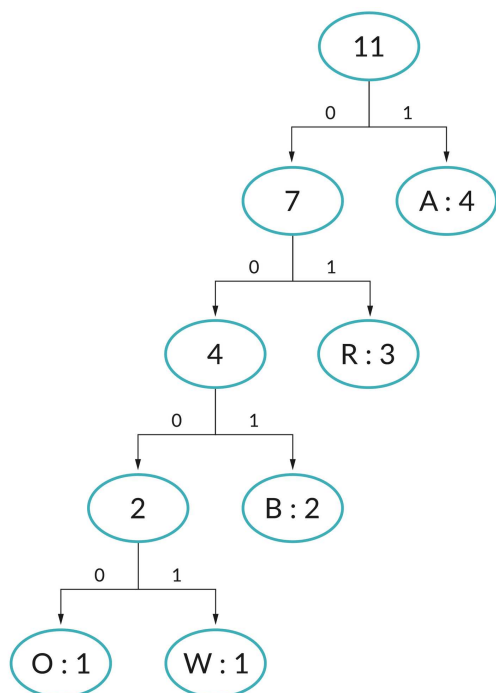
7

6

Ćwiczenie 6



Zaznacz symbol, który jest zapisany za pomocą najmniejszej liczby bitów na poniższym drzewie Huffmana.



☐ R

☐ O

☐ W

☐ B

☐ A

Ćwiczenie 7



Uporządkuj kolejne kroki kodowania słowa KOKI.

Utwórz listę trzech drzew binarnych, z nazwą symbolu K i liczbą wystąpień 2, z nazwą symbolu O i liczbą wystąpień 1, oraz z nazwą symbolu I i liczbą wystąpień 1.

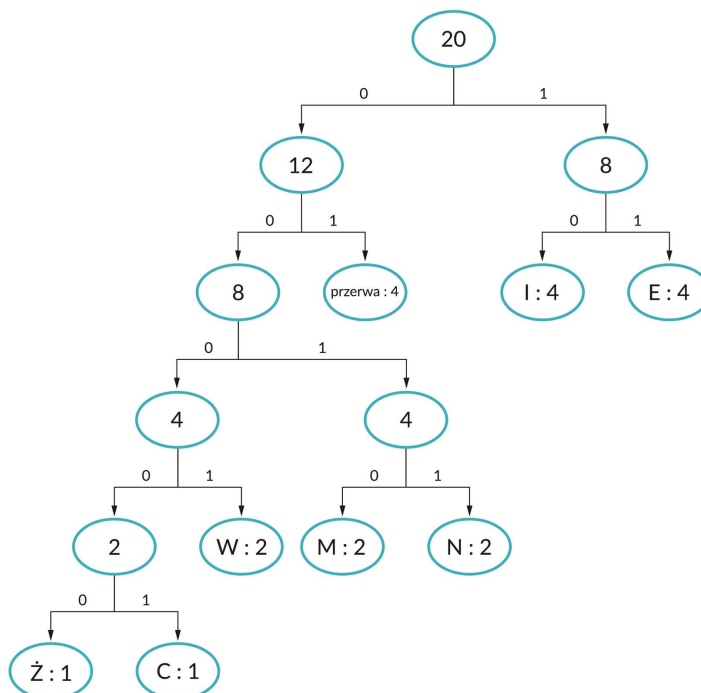
Połącz kody znaków tworząc zakodowane słowo KOKI.

Połącz dwa drzewa – z symbolem O oraz I. Będą one lewym i prawym poddrzewem tego drzewa. W korzeniu umieść sumę wystąpień, to znaczy wartość 2.

Połącz dwa drzewa – drzewo z symbolem K w korzeniu oraz drzewo z wartością 2 w korzeniu i symbolami O oraz I w poddrzewach. W korzeniu umieść wartość 4.

Utwórz kody znaków, przechodząc po krawędziach od korzenia do węzłów zawierających symbole.

Ćwiczenie 8



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Zakoduj znaną sentencję „WIEM ŻE NIC NIE WIEM” za pomocą drzewa Huffmana.

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Algorytm Huffmana

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

- I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.
- III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi, w tym: znajomość zasad działania urządzeń cyfrowych i sieci komputerowych oraz wykonywania obliczeń i programów.

Treści nauczania – wymagania szczegółowe

- I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

- 2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;
- 3) objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2) dokonuje kompresji informacji, objaśnia różnice między kompresją stratną i bezstratną tekstów, obrazów, dźwięków, filmów;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Dowiesz się, na czym polega algorytm Huffmana i poznasz listę czynności wykonywanych podczas jego realizacji.
- Przeanalizujesz kodowanie przykładowego wyrazu zgodnie z algorytmem Huffmana.
- Nauczysz się odkodowywać ciąg znaków, korzystając z drzewa Huffmana.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- telefony z dostępem do internetu.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Algorytm Huffmana”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Prowadzący wyświetla na tablicy interaktywnej zawartość sekcji „Wprowadzenie” i omawia cele do osiągnięcia w trakcie lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Uczniowie przystępują do cichego czytania tekstu e-materiału. Indywidualnie zapoznają się z treścią w sekcji „Przeczytaj”.
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Animacja”. Uczniowie zapoznają się z przykładem odkodowania ciągu bitów korzystając z drzewa Huffmana
3. **Ćwiczenie umiejętności.** Uczniowie realizują indywidualnie ćwiczenia nr 1-7, po ich wykonaniu porównują otrzymane wyniki z inną osobą.

Faza podsumowująca:

1. Nauczyciel zadaje pytania podsumowujące, np.
 - jaki rodzaj plików otrzymujemy po zastosowaniu kompresji bezstratnej?
 - czym jest drzewo binarne?
 - kod Huffmana jest kodem prefiksowym, co to oznacza?
2. Na koniec zajęć nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie wykonują ćwiczenie nr 8 z sekcji „Sprawdź się”.

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.

