

KODY HUFFMANA

Zadanie AiSD.L.Huff.01

Skonstruuuj optymalny kod Huffmana dla następujących częstości wystąpień znaków: a:3, d:5, e:6, o:8, p:8, r:9, u:12, t:12, v:13, z:20. Zakoduj przy pomocy utworzonego kodu słowo arvopart, a następnie je rozkoduj.

Zadanie AiSD.L.Huff.02

Udowodnij indukcyjnie (po liczbie liści), że koszt całego drzewa kodu Huffmana można obliczyć jako sumę, po wszystkich węzłach wewnętrznych, sum częstości wystąpień ich potomków.

UWAGA. Do implemetacji należy wybrać tylko jedno z poniższych zadań.

Zadanie AiSD.L.Huff.03

- (4 pkt.) Napisz program, który tworzy kod Huffmana dla podanego pliku. Każdy bajt pliku traktujemy jako znak (a więc zakres kodowanych znaków od 0 do 255). Program pobiera na wejściu plik, zlicza częstości wystąpień bajtów, tworzy kod Huffmana i drukuje tabelę kodów w postaci trójek:
 - znak (liczba $\in \{0, \dots, 255\}$, dla bajtów kodujących litery można drukować litery);
 - liczba wystąpień tego znaku;
 - kod Huffmana tego znaku.
- (1 pkt) Rozbuduj ww. program tak, żeby dodatkowo kodował wejściowy plik przy pomocy utworzonego kodu Huffmana tworząc nowy, zakodowany plik i następnie go rozkodowywał.
 - Do zakodowanego pliku nie trzeba dołączać tabeli kodów, co byłoby niezbędne w prawdziwym programie kodującym i rozkodującym.
 - Przy tworzeniu zakodowanego pliku należy pamiętać, że kody Huffmana są ciągami bitów, które trzeba upakowywać w bajty zakodowanego pliku. Rozwiązanie, w którym każdy bit kodu jest zapisywany znakowo jako bajt będzie punktowane na 0,5 punkta.

Uwaga 1. Tak jak napisano powyżej, można przyjąć, że każdy bajt czytanego pliku jest znakiem, czyli tekst jest w kodzie ASCII, ale też dopuszczamy pliki binarne do kodowania. Zatem na wejściu mamy znaki o kodach z zakresu 0-255, ale te które nie występują w kodowanym tekście po prostu pomijamy w tworzeniu kodu Huffmana.

Uwaga 2. Tekst, dla którego tworzymy kod Huffmana, czytamy z pliku, i może składać się on z wielu linijek. Program należy testować najpierw na małym pliku, ale też powinien działać na większym (patrz przykładowy plik 3700.txt).

Zadanie AiSD.L.Huff.04 (5 pkt.)

Napisz program, który pobiera na wejściu (z pliku) ciąg cyfr ze zbioru $\{0, 1, 2, 3, 4, 5, 6, 7\}$, zlicza częstości wystąpień cyfr, a następnie wyznacza, wypisuje i porównuje kodowania Huffmana uzyskane przez kodowanie po jednym znaku oraz kodowanie po parach znaków. Dokładniej mówiąc, wynik porównania to tabele przedstawiające długości kodów poszczególnych „znaków” (jednocyfrowych i dwucyfrowych) i długości zakodowanego tekstu (w bitach) w pierwszym i drugim przypadku (tekstu nie trzeba kodować, tylko policzyć długości jakie byłyby po zakodowaniu).