

Temat1 : Tworzenie słownika wspomagającego czytanie literatury obcej

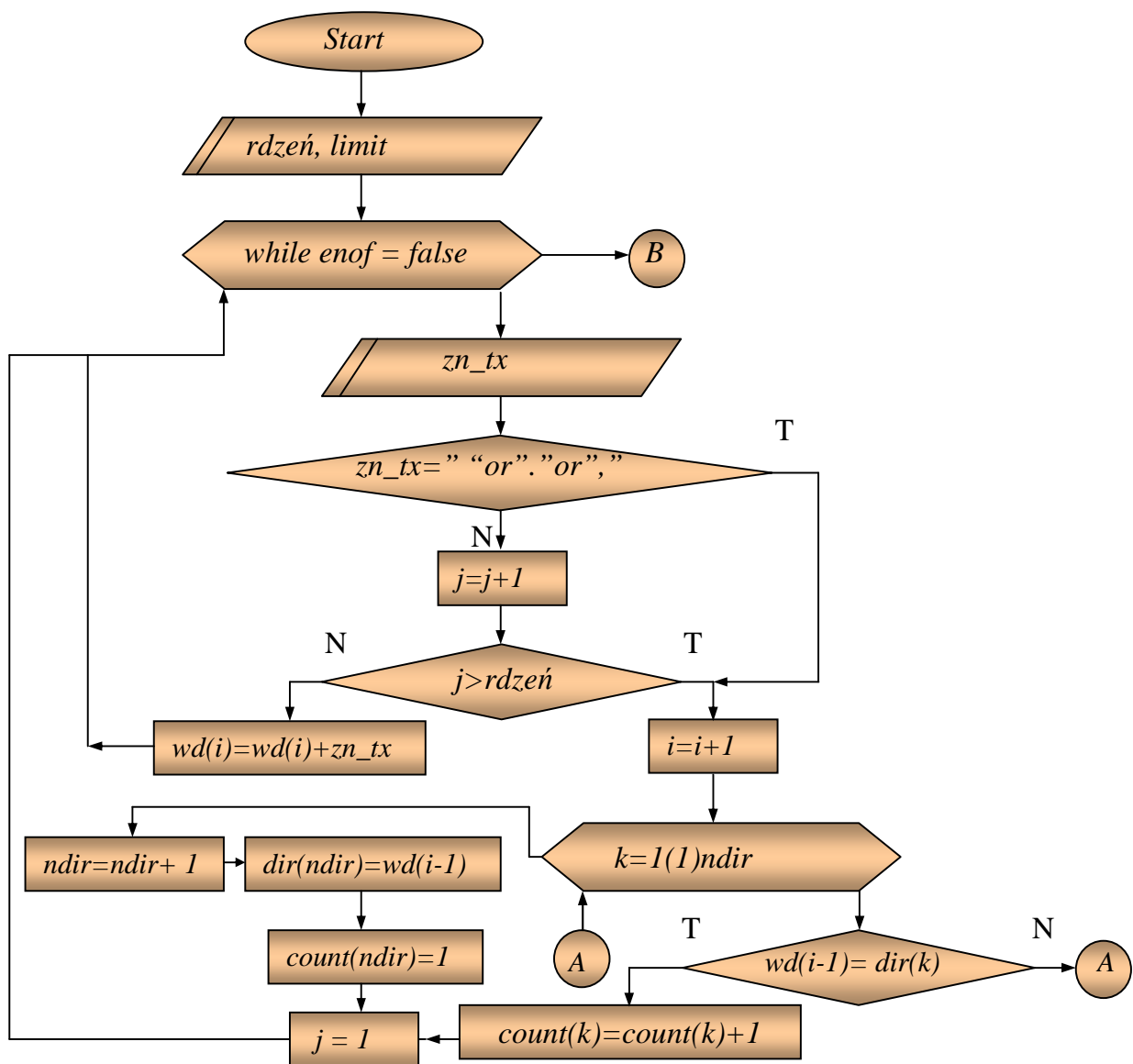
Cel: Kreowanie narzędzi wspomagania

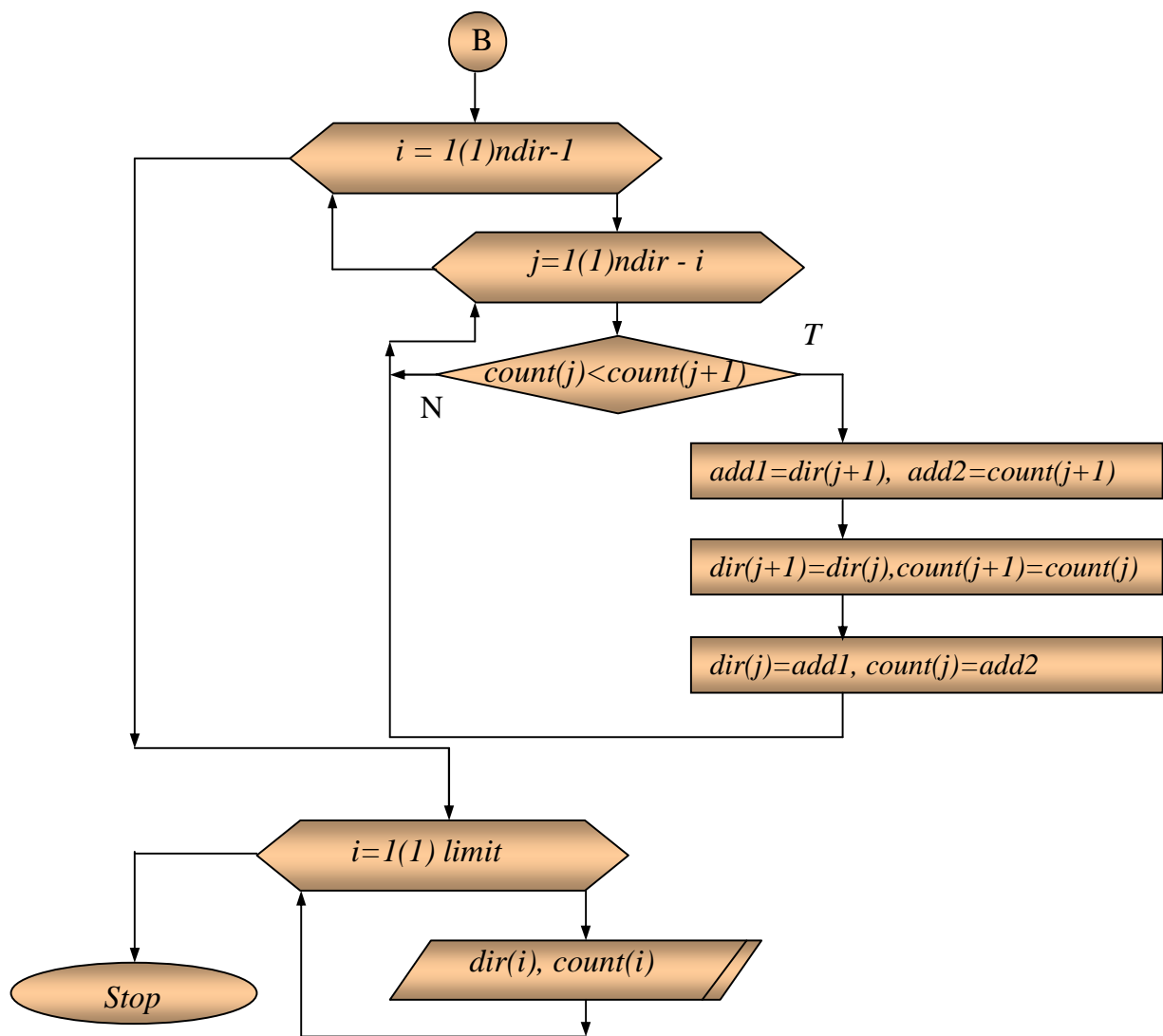
Metoda: Algorytmy wyszukiwania i sortowania

Opis metody :

- wybór słowa
- sprawdzenie niepowtarzalności słowa
- określenie częstości pojawienia się słowa

Schemat blokowy algorytmu





Opis schematu blokowego

Rozpoczynamy od określenia liczby słów, które możemy przyswoić (powtórzyć) przed rozpoczęciem czytania książki w języku obcym. Wielkość tą możemy nazwać limitem przyswajalności: *limit*. Zakładamy, iż dysponujemy plikiem (e-książką), który możemy poddać analizie. Książka zawiera słowa, które mieszczą się między spacją i kolejną spacją lub kropką, przecinkiem, myślnikiem, średnikiem. Każde słowo składa się z rdzenia, przedrostka, przyrostka. Przedrostek i rdzeń lub jego część tworzą „materiał słowotwórczy”: *rdzeń*. Materiał słowotwórczy będzie wykorzystany do tworzenia słownika wspomagającego czytanie książki. Kto chciałby bez problemu czytać książkę powinien w możliwie maksymalnym stopniu (*limit*) przyswoić (nauczyć się na pamięć) słów zebranych w słowniku. Niewielki procent literatury może być wyposażony w takie

słowniki. Wiemy jak kłopotliwe jeszcze w chwili obecnej jest korzystanie ze słowników nawet tych w postaci elektronicznej. Wygodnym rozwiązaniem byłyby książki elektroniczne gdzie po dotknięciu słowa pojawiałby się przetłumaczony odpowiednik ale takich książek jeszcze nie ma. Korzystanie ze słownika zawierającego wyrazy najczęściej występujące w książce jest, mimo dużej pracy przygotowawczej, najbardziej efektywne dla obecnych rozwiązań technicznych. Po wczytaniu wartości zmiennych *rdzeń* i *limit* rozpoczynamy czytanie pliku co będziemy powtarzali do momentu natknięcia się na znacznik końca pliku: *enof=true*. Czytanie realizujemy znakowo czyli znak po znaku: *zn_tx*. Po wczytaniu kolejnego znaku sprawdzamy znaczniki końca słów. Jeśli wczytany znak nie jest znacznikiem końca słowa sprawdzamy czy nie został utworzony „materiał słowotwórczy”: $j > rdzeń$. Materiał słowotwórczy zostanie więc utworzony przez zakończenie słowa lub przekroczenie długości rdzenia. Liczbę znaków w słowie podlicza licznik *j*, a liczbę analizowanych słów podlicza licznik *i*. Liczba nowych słów, a ściślej biorąc liczba materiałów słowotwórczych jest zawarta pod zmienną *ndir*. W algorytmie nie zainicjowano początkowej wartości *ndir* na poziomie jedności (co oczywiście należy zrobić na wstępie). Jeśli nowe słowo znajduje się w słowniku powiększamy licznik tego słowa: $count(k) = count(k) + 1$. Jeśli nowe słowo nie jest jeszcze umieszczone w słowniku zamieszczamy je tam: $dir(ndir) = wd(i-1)$, powiększając jednocześnie liczbę słów. Licznik częstości nowo umieszczonego w słowniku słowa ustawiamy na poziomie jedności: $count(ndir) = 1$. Po przeczytaniu całego pliku zawierającego książkę przystępujemy do porządkowania słownika według częstości występowania poszczególnych słów (sortowanie bąbelkowe od etykiety B). Z posortowanego zbioru wybieramy tylko limit słów, który będzie niezbędny (w założeniach) do zrozumienia treści książki.

Literatura

1. Banachowski L., Diks K., Rytter W.: Algorytmy i struktury danych, WNT, Warszawa, 1996
2. Cormen T.H., Leiserson Ch.E., Rivest R.L: Wprowadzenie do algorytmów, WNT, Warszawa, 1997
3. Sysło M.M., Deo N., Kowalik J.S.: Algorytmy optymalizacji dyskretniej, PWN, 1995
4. Boyer R.S., Moore J.S.: A fast string-searching algorithm, Communications of the ACM, 20(10), 1977