

ASD – zadanie programistyczne 2

Dane jest drzewo binarne etykietowane literami alfabetu angielskiego. Słowo w tym drzewie definiujemy jako ciąg etykiet kolejnych wierzchołków w drodze od liścia do korzenia. Znajdź i wypisz ostatnie alfabetycznie spośród takich słów.

Wejście

Ciąg wierszy opisujących wierzchołki. Każdy wiersz opisuje jeden wierzchołek i wygląda następująco:

X Y

gdzie:

X – etykieta wierzchołka

Y – ciąg liter L (left) i R (right) określających drogę od korzenia do danego wierzchołka

Przykład:

B LLR – oznacza, że wierzchołek do którego dostaniemy się idąc od korzenia w lewo (L), w lewo (L) i w prawo (R) ma etykietę B.

Wyjście

Ciąg liter tworzący ostatnie alfabetycznie słowo spośród wszystkich słów w danym drzewie.

Wymagania

- Złożoność czasowa średnia: $O(n \log n)$.
- Złożoność pamięciowa: $O(n)$.
- Program **musi** wczytywać dane wejściowe z pliku, którego lokalizacja podana jest w pierwszym parametrze wywołania programu.
- Jedyną rzeczą, którą program wypisuje, **musi** być linijka zawierająca odpowiedź.
- Całość rozwiązania musi zawierać się w **jednym** pliku.

Uwagi

- Ostateczny termin nadsyłania rozwiązań to 13.12.18 (piątek) do północy.
- Dozwolone języki to Java, C, C++ oraz C#.
- W przypadku rozwiązań w języku Java należy przestać **tylko i wyłącznie** plik sXXXXX.java, gdzie sXXXXX jest indeksem studenta.
- W przypadku rozwiązań w innych językach konieczne jest przesłanie kodu źródłowego **oraz** pliku wykonywalnego sXXXXX.exe.
- Zakładamy poprawność danych wejściowych.
- **Zabronione jest korzystanie z gotowych rozwiązań z języka Java (bądź innych), takich jak ArrayList, bądź StringBuilder. Wyjątkami są te rozwiązania, które służą do operacji wejścia/wyjścia (np. Scanner w Javie) oraz klasa String i metody na niej.**
- **Niezastosowanie się do opisanych w tym pliku wymagań i uwag skutkuje uzyskaniem 0 punktów, niezależnie od tego czy program działa poprawnie.**

Przykłady

1. Input

G RR

A

C L

F LLR

X LLL

F R

X RL

H LL

Output

XHCA

2. Input

B

C L

D LR

Output

DCB

3. Input

P LR

P RL

O RRR

O R

Z LRL

Y RLL

O L

T RR

W LLL

A LL

D

Z RLR

Output

ZPOD

4. Input

L L

A RL

K RR

K

C R

L RRL

Output

LKCK