

Algorytmy i Struktury Danych - Listy oraz złożoność obliczeniowa

Laboratorium nr 1 i 2 (4 x 45")

Grupa WCY20IY2S1

21.10.2021

Temat:

- a) *Złożoność obliczeniowa algorytmów. Pojęcie złożoności obliczeniowej: złożoność czasowa, złożoność pamięciowa. Asymptotyczna złożoność czasowa. Ocena złożoności obliczeniowej algorytmów iteracyjnych i rekurencyjnych.*
- b) *Listy. Rodzaje struktur listowych Podstawowe operacje na listach. Metody implementacji list.*

Pytanie

Wyjaśnij pojęcie listy jednokierunkowej, przedstaw strukturę elementu i podaj sposoby jej implementacji. Przedstaw zastosowanie listy jednokierunkowej. Odpowiedź zawrzyj w notatce do laboratorium.

Zad. nr 1

Pewien system generuje losowo w czasie rzeczywistym liczby naturalne w zakresie $<1, 2000>$. Zbuduj mechanizm przechwytywania liczb i liczby parzyste umieść na liście cyklicznej: A natomiast liczby nieparzyste na liście cyklicznej: B. Listy A i B są połączone a elementem łączącym staje się pierwszy, wygenerowany element, który nie należy do listy A lub B. Przechwytywane elementy są automatycznie układane w takie miejsca list A lub B, aby listy tworzyły zbiór elementów narastających (wg. wartości przechowywanej) zgodnie z kierunkiem wskazówek zegara.

Zbudowany program powinien posiadać zaimplementowane następujące funkcjonalności:

1. Funkcję **print_all(...)** – wypisującą wszystkie zawartość listy (wiersz: sąsiad lewy, wartość, sąsiad prawy lub sąsiad lewy, wartość, sąsiad prawy, węzeł);
2. Funkcję **print_range(int pcount, int dir, ...)** – która wypisuje pcount elementów z listy w kierunku dir=1 (zgodnie ze wskazówkami zegara), dir = -1 (przeciwnie);
3. Funkcję **pop(&adr)** – która usuwa z listy wskazany przez zmienną **adr** element;
4. Funkcję **push(..., val)** – która dodaje na koniec listy element;
5. Funkcję **insert(..., val, where)** – która wstawia do listy element przed element o adresie **where**;
6. Funkcję **dellist(...)** – która usuwa listę z pamięci;
7. Funkcję **printnode(...)** – wypisuje węzły łączące listy;
8. Funkcję **printlist(cnum)** – wypisuje zawartość **cnum** kręgu.

Uwagi dodatkowe:

- Kod programu należy umieścić w jednym pliku (włącznie z funkcjami) [pdf lub docx]. Przedstawione rozwiązanie powinno być wykonane samodzielnie. W przypadku stwierdzenia kopii wszyscy autorzy identycznych rozwiązań otrzymują ocenę ndst., co wiąże się z koniecznością poprawy ćwiczenia w terminie konsultacji.

- Do implementacji proszę użyć języka C w standardzie ANSI C lub C++ (bez typów generycznych i bibliotek specjalistycznych)
- Do ćwiczenia laboratoryjnego sporządzamy notatkę w formie elektronicznej, która jest dodawana do ćwiczenia w oddzielnym pliku w formacie {docx, pdf, odt}. W notatce zamieszczamy odpowiedzi na pytania opisowe i niezbędne obliczenia.
- Sprawdzenie pracy następuje w czasie zajęć laboratoryjnych, po zgłoszeniu przez studenta gotowości do sprawdzenia pracy.
- Gotowe rozwiązania (sprawozdanie i kod programu – dwa pliki niespakowane) przesyłamy za pomocą narzędzia MS Teams do oceny. Ocena po sprawdzeniu będzie dostępna dla każdego studenta w systemie MS Teams.