Instytut Informatyki UWr

Wstęp do informatyki

Lista 10

C: Typ `struct elem` zdefiniowany jest następująco:

```
struct elem {
     int val;
     struct elem *next;
     };
```

W rozwiązaniach zadań można korzystać z funkcji

```
struct elem *utworz(int wart)
```

tworzącej listę jednoelementową z kluczem `wart` (p. slajdy do wykładu).

Python: Listy wiązane tworzymy przy pomocy klasy ListItem z polami `val` i `next` oraz następującym konstruktorem:

```
class ListItem:
   def __init__(self,value):
      self.val = value
      self.next = None
```

A zatem listę jednoelementową z kluczem `wart` tworzymy tak:

```
ListItem(wart)
```

Zadania:

Napisz funkcje realizujące następujące operacje:

1. [0.5] Dołączenie nowego elementu na koniec listy.
2. [0.5] Usunięcie ostatniego elementu z listy.
3. [1] Dołączenie jednej listy na koniec drugiej.
4. [1] Usunięcie z listy **wszystkich** elementów o podanej wartości pola `val`. Złożoność czasowa Twojego rozwiązania powinna być $O(n)$, gdzie n to liczba elementów na liście.

5. [1] Zaproponuj sposób reprezentacji listy (jednokierunkowej), który umożliwi realizację operacji z zadań 1 i 3 oraz operacji wstawiania/usuwania elementu z początku listy w czasie $O(1)$. Podaj jak zaimplementować te operacje w nowej reprezentacji.

   Uwaga. Dodatkowy koszt pamięciowy związany z nową reprezentacją powinien być $O(1)$.
6. [1] Napisz funkcję wypisującą na standardowym wyjściu elementy listy w odwrotnej kolejności do ich występowania w liście. Nie należy przy tym zmieniać kolejności elementów w liście ani tworzyć nowej listy.
   Wskazówka. Wykorzystaj następującą obserwację: aby wypisać od końca elementy niepustej listy L, wystarczy najpierw wypisać od końca elementy L bez pierwszego elementu, a potem wypisać pierwszy element L.
7. [1] Napisz funkcję umożliwiającą odwrócenie kolejności elementów na liście jednokierunkowej. W Twojej implementacji nie powinny być tworzone nowe elementy listy a jedynie zmieniane wskaźniki.

8. [1] Napisz funkcję, która rozdzieli daną listę na dwie podlisty: jedną zawierającą elementy z kluczami dodatnimi a drugą – elementy z kluczami ujemnymi. W Twojej implementacji nie powinny być tworzone nowe elementy listy a jedynie zmieniane wskaźniki.
9. [2] Zaproponuj typ danych dla elementów listy dwukierunkowej, tj. takiej, w której każdy element zawiera wskaźnik na następny i wskaźnik na poprzedni element w liście (elementy pierwszy i ostatni mają odpowiednie wskaźniki ustawione na NULL/None). Napisz funkcje realizujące operacje kolejkowe na takiej liście (dodanie elementu na koniec kolejki, usunięcie elementu z początku kolejki) a także usunięcie elementu z końca listy.
10. [1] Napisz funkcję, która scala dwie **uporządkowane** listy dając również w wyniku listę uporządkowaną. W Twojej implementacji nie powinny być tworzone nowe elementy listy a jedynie zmieniane wskaźniki.

Zadania dodatkowe, nieobowiązkowe (nie wliczają się do puli punktów do zdobycia na ćwiczeniach, punktacja została podana tylko jako informacja o trudności zadań wg wykładowcy)

11. [0.5] Napisz funkcję wypisującą na standardowym wyjściu wszystkie dodatnie elementy z listy.
12. [0,5] Przedstaw sposób implementacji stosu przy pomocy list, czyli podaj funkcje realizujące operacje: wstaw element na szczyt stosu, usuń element ze szczytu stosu, zainicjuj (pusty) stos.
13. [0,5] Przedstaw sposób implementacji kolejki przy pomocy list, czyli podaj funkcje realizujące operacje: wstaw element na koniec kolejki, usuń element z początku kolejki, zainicjuj (pustą) kolejkę.
14. [1] Napisz funkcje pozwalające usuwać i dodawać elementy do **uporządkowanej** listy dwukierunkowej.