

Egzaminy i inne zadania. Semestr II.

Poniższe zadania są wyborem zadań ze Wstępu do Informatyki z egzaminów jakie przeprowadziłem w ciągu ostatnich lat. Ponadto dołączyłem szereg zadań, które pojawiały się na różnych sprawdzianach i przy innych okazjach w tym okresie. Zadania obejmują materiał drugiego semestru.

Marek Zawadowski

Egzamin ze Wstępu do Informatyki. 10 września 2019.

Zadanie 1

Dana jest lista jednokierunkowa liczb całkowitych. Napisz funkcję, która w miejsce pierwszej liczby ujemnej na tej liście wpisze sumę wszystkich wyrazów ujemnych. Pozostałe ujemne elementy listy należy usunąć z listy (i zniszczyć). Jeśli na liście nie ma liczb ujemnych, to funkcja ma zwrócić listę niezmienną.

Przykład. Listę 2, -4, 56, -7, 7, -3, 8 funkcja powinna przekształcić w listę 2, -14, 56, 7, 8.

Zadanie 2

Luka w drzewie BST o kluczach całkowitych jest para kluczy $\langle k1, k2 \rangle$ tego drzewa ($k1 \leq k2$), pomiędzy którymi nie ma żadnych kluczy w tym drzewie. Długość luki $\langle k1, k2 \rangle$ to $k2 - k1$. Napisz funkcję, która dla danego drzewa BST zwraca długość najdłuższej luki w tym drzewie.

Zadanie 3

Napisz funkcję, która sprawdzi czy graf skierowany reprezentowany przez listy incydencji jest drzewem.

Zadanie 4

Sformułuj problem sortowania topologicznego. Opisz algorytm rozwiązujący ten problem. Jaka jest złożoność tego algorytmu? Opisz działanie tego algorytmu dla grafu reprezentowanego przez następujące listy incydencji:

```
1 : 2 5 6
2 : 3
3 : 4
4 :
5 : 4
6 : 5
7 : 4
8 : 4
9 : 8
10 : 4 8
```

Egzamin ze Wstępu do Informatyki. 24 czerwca 2019.

Zadanie 1

Dane są dwie listy liczb całkowitych uporządkowane ściśle rosnąco. Napisz funkcję, która przeniesie powtarzające się elementy na tych listach z pierwszej listy na trzecią, a z drugiej je usunie. Dla list $l1 = [1, 3, 5, 7]$, $l2 = [3, 4, 5, 6]$, funkcja ma zwrócić listy $l1 = [1, 7]$, $l2 = [4, 6]$, $l3 = [3, 5]$.

Zadanie 2

Wierzchołek v drzewa binarnego jest *stabilny*, gdy każda droga z v do liścia ma tę samą długość. Drzewo jest *stabilne*, jeśli każdy jego wierzchołek jest stabilny. Napisz funkcję sprawdzającą, czy dane drzewo binarne jest stabilne.

Zadanie 3

Źródło s to wierzchołek grafu skierowanego, z którego prowadzi droga do każdego innego wierzchołka tego grafu. Napisz procedurę, która dla grafu skierowanego $G = (V, E)$ reprezentowanego jako listy incydencji, wierzchołek v i źródła s , zwróci takie źródło w grafie G (być może różne od s), z którego droga do v jest najkrótsza. Można zakładać, że mamy zaimplementowane funkcje obsługujące kolejkę: `int pop()`, `void push(int v)`, `int empty()`.

Rozwiązania o złożoności większej niż $O(|V| + |E|)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 4

Zdefiniuj, co to jest minimalne drzewo rozpinające dla grafu niezorientowanego z wagami. Dla jakich grafów niezorientowanych istnieją minimalne drzewa rozpinające? Opisz algorytm znajdujący minimalne drzewa rozpinające dla grafów, dla których takie drzewa istnieją. Jaka jest złożoność tego algorytmu?

Naszkicuj minimalne drzewo rozpinające dla grafu G z funkcją wag w reprezentowanych przez następujące listy incydencji:

```
1 : (2, 2) (4, 6)
2 : (1, 2) (3, 4) (6, 3) (5, 2)
3 : (2, 4) (6, 2)
4 : (1, 6) (5, 4) (8, 6)
5 : (2, 2) (7, 1) (4, 4) (9, 3)
6 : (3, 2) (2, 3) (7, 6)
7 : (6, 6) (5, 1) (9, 7)
8 : (4, 6) (9, 4)
9 : (5, 3) (7, 7) (8, 4)
```

Para na liście oznacza koniec krawędzi i jej wagę. Na przykład druga para na pierwszej liście (4,6) oznacza krawędź z 1 do 4 z wagą 6.

Kolokwium ze Wstępu do Informatyki. 23 kwietnia 2018.

Zadanie 5

Dana jest lista 1-kierunkowa liczb całkowitych i liczba $k > 0$. Napisz funkcję, która o ile istnieje k -ty element na liście i jest różny od zera, to go z tej listy usunie i zwróci jego wartość a w przeciwnym przypadku zwróci 0.

Przykład. Dla $k = 3$ oraz listy 2, 0, 5, 4, funkcja powinna zwrócić 5 i listę 2, 0, 4.

Zadanie 6

Dane jest drzewo binarne liczb całkowitych. Węzeł drzewa nazywamy p-dobrym, jeśli w poddrzewie zaczynającym się od tego węzła jest więcej elementów parzystych niż nieparzystych. Napisz funkcję, która policzy, ile jest węzłów p-dobrych w danym drzewie binarnym.

Zadanie 7

Co to jest drzewo BST? Co to jest drzewo czerwono-czarne? Podaj przykład drzewa czerwono-czarnego o kluczach: 8, 12, 13, 14, 20, 21, 22, 28, 29, 31. Jaki jest koszt włożenia elementu na drzewo BST, a jaki na drzewo czerwono-czarne?

Egzamin Poprawkowy ze Wstępu do Informatyki. 6 września 2018.

Zadanie 1

Napisz funkcję, która dla listy jednokierunkowej liczb całkowitych zwróci sumę dodatnich elementów tej listy. Dla listy $12, -6, -9, 2, -3$ funkcja powinna zwrócić liczbę 14.

Zadanie 2

Napisz procedurę, która z drzewa BST o kluczach całkowitych usunie wszystkie liście o parzystych kluczach.

Zadanie 3

Dany jest graf nieorientowany spójny $G = (V, E)$, $V = \{1, \dots, n\}$ reprezentowany przez listy incydencji oraz wierzchołek $x \in V$. Napisz procedurę, która sprawdzi czy po usunięciu x i krawędzi z incydentnych x z grafu G , tak powstały graf jest nadal spójny.

Rozwiązania o złożoności powyżej $O(|V| + |E|)$ będą ocenian w skali od 0 do 8 punktów.

Zadanie 4

Sformułuj problem sortowania topologicznego. Opisz algorytm sortowania topologicznego. Jaka jest złożoność tego algorytmu? W jakim porządku ustawi algorytm wierzchołki grafu reprezentowane przez listy incydencji

1 : 3, 4

2 : 1, 6

3 : 4

4 :

5 : 7

6 : 3, 5

7 : 4

8 : 7

Egzamin ze Wstępu do Informatyki. 18 czerwca 2018.

Zadanie 1

Napisz procedurę, która z listy jednokierunkowej liczb całkowitych usunie liczby, które nie są względnie pierwsze z wszystkimi poprzednimi (jeszcze nie usuniętymi). Dla listy $2, 6, 9, 10, 33, 35, 143$ procedura powinna zwrócić listę $2, 9, 35, 143$.

Zadanie 2

Drzewo BST jest *prawe* o ile prawe poddrzewo każdego wierzchołka tego drzewa ma nie mniej węzłów niż lewe poddrzewo tego wierzchołka. Napisz funkcję, która sprawdzi czy dane drzewo BST jest prawe.

Zadanie 3

Dany jest graf skierowany $G = (V, E)$, $V = \{1, \dots, n\}$ reprezentowany przez listy incydencji oraz dwa wierzchołki $x, y \in V$. Napisz procedurę, która znajdzie wierzchołek $a \in V$ taki, że łączna długość dróg z a do x i z a do y jest najmniejsza. Rozwiązania o złożoności powyżej $O(|V| + |E|)$ będą ocenian w skali od 0 do 8 punktów.

Zadanie 4

Co to jest silnie spójna składowa grafu zorientowanego? Sformułuj problem znajdowania silnie spójnych składowych. Opisz algorytm rozwiązujący ten problem. Jaka jest złożoność tego algorytmu. Opisz działanie tego algorytmu na przykładzie grafu:

1 :	2	6	5
2 :	3	7	
3 :	4		
4 :	1	5	
5 :	9	10	
6 :	7		
7 :	6	8	
8 :	5	9	
9 :	10		
10 :			

Kolokwium ze Wstępu do Informatyki. 23 kwietnia 2018.

Zadanie 1

Napisz funkcję, która dla drzewa BST zwróci liczbe jego wierzchołków, które mają dokładnie jednego syna.

Zadanie 2

Napisz procedurę, która dla dwóch list ściśle rosnących liczb dodatnich i liczby całkowitej n niemniejszej niż wszystkie elementy na obu listach, zwróci ‘różnicę symetryczną’ tych list, to znaczy listę ściśle rosnącą, która jest stworzona z liczb całkowitych z przedziału od 1 do n , które są albo na obu listach albo na żadnej z nich. Dla list $l1 = [2, 3, 5, 7]$, $l2 = [3, 4, 6, 7, 8]$ i liczby $n = 10$ procedura powinna zwrócić listę $l = [1, 3, 7, 9, 10]$, usuwając zbędne elementy.

Zadanie 3

Co to jest drzewo BST? Co to jest drzewo czerwono-czarne? Podaj przykład drzewa czerwono-czarnego o kluczach: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. Jaki jest koszt włożenia elementu na drzewo BST a jaki na drzewo czerwono-czarne?

Egzamin ze Wstępu do Informatyki. 30 sierpnia 2017.

Zadanie 1

Napisz procedurę, która mając dane dwie listy jednokierunkowe $l1$ i $l2$ liczb całkowitych uporządkowane nierosnąco scali je w jedną listę uporządkowaną niemalejąco. Dla list $l1 : 7, 6, 3, 2, 2$ i $l2 : 12, 8, 6, 2$ procedura powinna zwrócić listę 2, 2, 2, 3, 6, 6, 7, 8, 12.

Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 2

Defekt wierzchołka v o wysokości h , to jeden minus iloraz liczby wierzchołków w poddrzewie o wysokości h i $2^{(h+1)} - 1$ (maksymalnej liczby liczb wierzchołków w drzewie o wysokości h), tzn. $Dt(v) = 1 - \frac{\text{rozmiar}(v)}{2^{(\text{wysokosc}(v)+1)} - 1}$. Napisać procedurę, która zwraca wskaźnik do wierzchołka o największym defekcie.

Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 3

Napisz procedurę, która mając graf skierowany G reprezentowany przez listy incydencji i wierzchołek v tego grafu, zwróci liczbę wierzchołków do których można dojść z v w grafie G .

Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 4

Opisz działanie algorytmu przeszukiwania grafu w głąb (DFS) na przykładzie grafu

```
1 : 2 5
2 : 1 6
3 : 2 4 7
4 : 8
5 : 6
6 :
7 : 6
8 : 7
```

Jaka jest złożoność tego algorytmu? Podaj dwa przykłady zastosowania tego algorytmu. Opisz jeden z nich dokładnie.

Egzamin ze Wstępu do Informatyki. 19 czerwca 2017.

Zadanie 1

Napisz procedurę, która mając dane dwie listy jednokierunkowe $l1$ i $l2$ liczb całkowitych uporządkowane rosnąco usunie z listy $l2$ te elementy, których NIE MA w $l1$ i wstawi je do $l1$ z zachowaniem uporządkowania. Dla list $l1 : 2, 3, 5, 7$ i $l2 : 2, 4, 6, 7, 8$ procedura powinna zwrócić listy $l1 : 2, 3, 4, 5, 6, 7, 8$ i $l2 : 2, 7$.

Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 2

Wierzchołek drzewa BST jest zbalansowany, jeśli wysokości podrzew (lewego i prawego) różnią się co najwyżej o 1. Drzewo jest chaotyczne, jeśli ma więcej wierzchołków niezbalansowanych niż zbalansowanych. Napisz procedurę, która zwróci wskaźnik do korzenia największego (w sensie rozmiaru) chaotycznego poddrzewa.

Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 3

Cykl $x_0, \dots, x_n$ grafu skierowanego $G = (V, E)$, gdzie $V = \{1, \dots, n\}$, jest *cyklem Hamiltona*, jeśli $x_0 = x_n$ oraz wierzchołki $x_1, \dots, x_n$ są różne. Napisz procedurę, która mając graf skierowany i listę wierzchołków tego grafu, sprawdzi czy wierzchołki na liście tworzą cykl Hamiltona.

Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 4

Sformułuj problem znajdowania silnie spójnych składowych grafu zorientowanego. Opisz algorytm znajdujący silnie spójne składowe grafu zorientowanego. Jaka jest złożoność tego algorytmu? Opisz jego działanie dla grafu

```

1 : 2 6
2 : 1 5
3 : 2 4 8
4 : 7
5 : 6
6 : 5
7 : 4
8 : 3

```

Egzamin ze Wstępu do Informatyki. 29 września 2016.

Zadanie 1

Lista jednokierunkowa powstała przez dołączenie na koniec listy rosnącej drugiej listy malejącej. Napisz procedurę, która posortuje niemalejąco tę listę. Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 2

Wierzchołek w drzewie binarnym jest *zrównoważony*, jeśli rozmiar jego poddrzew, lewego i prawego, jest równy. Napisz procedurę, która policzy liczbę zrównoważonych wierzchołków w drzewie binarnym. *Rozmiar drzewa* to liczba wierzchołków drzewa.

Zadanie 3

Opisz algorytm, który sprawdzi, czy w grafie skierowanym istnieje taki wierzchołek, z którego są osiągalne wszystkie wierzchołki.

Zadanie 4

Sformułuj problem znajdowania silnie spójnych składowych grafu skierowanego. Opisz algorytm rozwiązujący ten problem. Jaka jest złożoność tego algorytmu? Opisz działanie tego algorytmu dla grafu reprezentowanego przez listy incydencji:

```

1 : 2
2 : 8 5
3 : 9
4 : 1 2
5 : 6
6 : 5 4 5
7 : 3
8 : 1 2 4 6
9 : 10
10 : 3

```

Egzamin ze Wstępu do Informatyki. 31 sierpnia 2016.

Zadanie 1

Napisz funkcję, która - mając listę uporządkowaną liczb całkowitych dodatnich - stworzy i zwróci listę liczb całkowitych dodatnich, których nie ma na pierwszej liście i które nie są większe od największej liczby na pierwszej liście. Dla pustej listy wejściowej należy zwrócić pustą listę. Na przykład, dla listy 2 3 6 9 należy stworzyć i zwrócić listę 1 4 5 7 8

Zadanie 2

Napisz funkcję, która sprawdzi, czy w drzewie binarnym istnieje taki węzeł, że w jego lewym niepustym poddrzewie jest tyle samo węzłów co w prawym.

Zadanie 3

Napisz funkcję, która sprawdzi, czy dany wierzchołek v w grafie skierowanym reprezentowanym przy pomocy list incydencji leży na pewnym cyklu tego grafu.

Zadanie 4

Sformułuj problem sortowania topologicznego grafu skierowanego. Opisz algorytm sortujący topologicznie graf skierowany. Jakie grafy skierowane można posortować topologicznie? Jaka jest złożoność algorytmu sortowania topologicznego? Jak będzie wyglądał efekt wykonania algorytmu sortowania topologicznego na grafie skierowanym reprezentowanym przez następujące listy incydencji:

```
1 : 3 5 6
2 : 3
3 :
4 : 1 2
5 :
6 :
7 : 6
8 : 3
```

Egzamin ze Wstępu do Informatyki. 22 czerwca 2016.

Zadanie 1

Dana jest lista napisów (typu `string`). Napisać funkcję, która zwróci napis z tej listy, który zawiera najwięcej cyfr. Jeśli jest więcej niż jeden taki napis, to ten, który jest najdłuższy. Jeśli i takich jest więcej niż jeden, to najwcześniejszy na liście.

Uwaga. Funkcja `length` zwraca długość stringu. Jeśli `s` jest zmienna typu `string`, to `length(s)` zwraca długość `s` a `s[i]` zwraca i -ty charakter stringu `s` dla $i = 1, \dots, \text{length}(s)$. $\text{length}('ala') = 3$, $'ala'[2] = 'l'$.

Zadanie 2

Współczynnik zagęszczenia drzewa to stosunek liczby węzłów do wysokości drzewa. Napisać funkcję, która dla danego drzewa binarnego zwróci wskaźnik do wierzchołka jego poddrzewa o największym współczynniku zagęszczenia.

Zadanie 3

Napisz funkcję, która dla grafu skierowanego $G = (V, E)$ reprezentowanego jako tablica list incydencji oraz wierzchołka $v \in V$ zwraca `true`, jeśli do v można dojść z każdego innego wierzchołka i `false` w przeciwnym przypadku.

Zadanie 4

Sformułuj problem sortowania topologicznego grafu zorientowanego. Opisz algorytm rozwiązujący ten problem. Jaka jest złożoność tego algorytmu? Opisz jego działanie dla grafu

```
1 : 2 5
2 : 3
3 : 7
4 : 3 6 8
5 : 3
6 : 7
7 :
8 :
```

Egzamin ze Wstępu do Informatyki. 2 września 2015.

Zadanie 1

Napisz procedurę, która poprzestawia wyrazy listy jednokierunkowej liczb całkowitych tak by wszystkie liczby parzyste stały przed wszystkimi liczbami nieparzystymi.

Na przykład dla listy: 2, 3, 4, 7, 5, 18 5, 7, 8, 9 procedura może zwrócić listę 2, 4, 18, 8, 3, 7, 5, 5, 7, 9.

Zadanie 2

Drzewo binarne jest *zbalansowane względem liści*, jeśli dla każdego wierzchołka tego drzewa różnica liczby liści jego poddrzew, lewego i prawego, wynosi co najwyżej 1. Napisz pocedurę, która sprawdzi, czy dane drzewo binarne jest zbalansowane względem liści.

Zadanie 3

Wierzchołek v jest *punktem artykulacji* nieskierowanego grafu spójnego $G = (V, E)$, jeśli po usunięciu z grafu G wierzchołka v i wszystkich krawędzi incydentnych z v powstaje graf niespójny. Napisz funkcję, która dla grafu nieskierowanego spójnego $G = (V, E)$ ($V = \{1, \dots, n\}$) reprezentowanego jako tablica list incydencji oraz wierzchołka $1 \in V$, zwraca **true** jeśli 1 jest punktem artykulacji i **false** w przeciwnym przypadku.

Zadanie 4

Sformułuj problem znajdowania silnie spójnych składowych grafu zorientowanego. Opisz algorytm rozwiązujący ten problem. Jaka jest złożoność tego algorytmu? Opisz jego działanie dla grafu

```
1 : 2 5
2 : 3
3 : 2 4 7
4 :
5 : 1 6
6 : 3
7 : 4 6 7
8 : 4
```

Egzamin ze Wstępu do Informatyki. 23 czerwca 2015.

Zadanie 1

Napisz procedurę, która usunie z listy jednokierunkowej liczb całkowitych liczby nieparzyste.

Na przykład dla listy: 2, 3, 4, 7, 5, 18 5, 7, 8, 9 procedura powinna zwrócić listę 2, 4, 18, 8.

Zadanie 2

Drzewo binarne jest *zbalansowane* jeśli dla każdego wierzchołka tego drzewa różnica rozmiarów jego poddrzew, lewego i prawego, jest co najwyżej 1. Napisz pocedurę, która sprawdzi czy dane drzewo binarne jest zbalansowane.

Zadanie 3

Napisz funkcję, która dla grafu skierowanego $G = (V, E)$ reprezentowanego jako tablica list incydencji oraz wierzchołka $v \in V$, zwraca **true** jeśli z v można dojść do każdego innego wierzchołka i **false** w przeciwnym przypadku.

Zadanie 4

Sformułuj problem znajdowania silnie spójnych składowych grafu zorientowanego. Opisz algorytm rozwiązujący ten problem. Jaka jest złożoność tego algorytmu? Opisz jego działanie dla grafu

```
1 : 2 5
2 : 3
3 : 2 4 7
4 :
5 : 1 6
6 : 3
7 : 4 6 7
8 : 4
```

Egzamin Poprawkowy ze Wstępu do Informatyki. 11 września 2014.

Zadanie 1

Dane są dwie listy jednokierunkowe liczb całkowitych uporządkowane niemalejąco. Napisz procedurę, która usunie te elementy, które występują na obu listach.

Przykład. Procedura mając na wejściu listy

```
1 2 2 3 4 5   i   0 1 1 3 3 6
```

powinna zwrócić listy

```
0 2 2 4 5   i   0 6
```

Zadanie 2

Napisz funkcję, która dla drzewa binarnego etykietowanego liczbami całkowitymi obliczy liczbę liści, do których można dojść od korzenia wyłącznie po wierzchołkach o parzystych etykietach.

Zadanie 3

Opisz algorytm, który obliczy długość najdłuższej ścieżki (prostej) w grafie skierowanym acyklicznym.

Zadanie 4

Opisz algorytm przeszukiwania grafu w głąb. Jaka jest jego złożoność? Jakie znasz algorytmy używające tego algorytmu? Opisz jeden z nich.

Egzamin ze Wstępu do Informatyki. 13 czerwca 2014.

Zadanie 1

Lista jednokierunkowa powstała przez dołączenie na koniec listy rosnącej drugiej listy malejącej. Napisz procedurę, która posortuje tę listę. Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 2

Wierzchołek w drzewie binarnym jest *prawy*, jeśli jego prawe poddrzewo ma rozmiar nie mniejszy od lewego poddrzewa. Napisz procedurę, która policzy liczbę prawych wierzchołków w drzewie binarnym. *Rozmiar drzewa* to liczba wierzchołków drzewa.

Zadanie 3

Opisz algorytm, który sprawdzi, czy w grafie skierowanym istnieje taka para wierzchołków, że każdy wierzchołek tego grafu jest osiągalny z co najmniej jednego z nich.

Zadanie 4

Sformułuj problem znajdowania silnie spójnych składowych grafu skierowanego. Opisz algorytm rozwiązujący ten problem. Jaka jest złożoność tego algorytmu? Opisz działanie tego algorytmu dla grafu reprezentowanego przez listy incydencji:

```
1 : 2
2 : 8 5
3 : 9
4 : 1 2
5 : 6
6 : 5 4 5
7 : 3
8 : 1 2 4 6
9 : 10
10 : 3
```

Egzamin Poprawkowy ze Wstępu do Informatyki 2. 12 września 2013.

Zadanie 1

Lista dwukierunkowa z wartownikiem liczb całkowitych uporządkowana rosnąco została przesunięta cyklicznie o pewną liczbę miejsc. Napisz procedurę, która tak przestawi elementy tej listy, by znowu była uporządkowana rosnąco. *Uwaga.* Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali od 0 do 8 punktów.

Zadanie 2

Napisz funkcję, która dla danego drzewa binarnego zwróci liczbę wierzchołków tego drzewa, które są korzeniami pełnych poddrzew binarnych.

Zadanie 3

Wierzchołek grafu zorientowanego jest minimalny, jeśli można do niego dojść wyłącznie z wierzchołków tej samej silnie spójnej składowej. Opisz algorytm, który znajduje wszystkie wierzchołki minimalne grafu. *Uwaga.* Algorytmu nie trzeba zapisywać w języku Pascal, wystarczy opisać go słowami.

Zadanie 4

Zdefiniuj, co to jest minimalne drzewo rozpinające dla grafu niezorientowanego z wagami. Dla jakich grafów niezorientowanych istnieją minimalne drzewa rozpinające? Opisz algorytm znajdujący minimalne drzewa rozpinające dla grafów, dla których takie drzewa istnieją. Jaka jest złożoność tego algorytmu?

Egzamin ze Wstępu do Informatyki 2. 11 czerwca 2013.

Zadanie 1

Dana jest lista dwukierunkowa z wartownikiem liczb całkowitych. Napisz procedurę, która zwróci najdłuższy jej fragment uporządkowany niemalejąco a resztę elementów usunie. Dla listy 2 3 1 2 4 5 3 4 6 6 5 procedura powinna zwrócić albo fragment 1 2 4 5 albo 3 4 6 6.

Zadanie 2

Napisz funkcję, która dla danego drzewa binarnego zwróci wskaźnik do takiego liścia tego drzewa, że na ścieżce od korzenia do tego liścia jest największa liczba prawych synów.

Zadanie 3

Opisz algorytm, który dla danego grafu skierowanego G zwróci listę wszystkich wierzchołków tego grafu, z których są osiągalne wszystkie wierzchołki tego grafu. Jeśli nie ma takich wierzchołków, zwracana lista powinna być pusta. *Uwaga.* Algorytmu nie trzeba zapisywać w języku Pascal, wystarczy opisać go słowami.

Zadanie 4

Co to są silnie spójne składowe grafu skierowanego? Opisz algorytm znajdowania silnie spójnych składowych grafu skierowanego. Jaka jest złożoność tego algorytmu? Opisz działanie tego algorytmu dla grafu, którego reprezentacja przez listy incydencji wygląda tak

```
1 : 2 5
2 : 1
3 : 6
4 : 3 8
5 :
6 : 2 4 5
7 : 3 5 6 8
8 : 7
```

**Egzamin Poprawkowy ze Wstępu do Informatyki.
13 września 2012.**

Zadanie 1

Dana jest lista liczb całkowitych posortowana niemalejąco. Napisać procedurę, która usunie z tej listy elementy powtarzające się i wstawi je na nową listę, też uporządkowaną niemalejąco. Z listy 1, 3, 3, 3, 4, 5, 5 procedura powinna stworzyć dwie listy 1, 3, 4, 5 i 3, 3, 5.

Zadanie 2

Napisać funkcję, która dla danego drzewa binarnego zwraca korzeń poddrzewa o minimalnej wysokości, które nie jest drzewem BST, lub `nil` gdy całe drzewo jest drzewem BST.

Zadanie 3

Dany jest graf skierowany. Opisać algorytm który sprawdza czy graf jest acykliczny. W przypadku gdy graf nie jest acykliczny zwraca jeden z cykli grafu. *UWAGA.* To zadanie można rozwiązać opisując algorytm słownie (tzn. nie jako procedurę w języku Pascal) używając do opisu algorytmów poznanych na wykładzie.

Zadanie 4

Co to jest składowa spójna grafu nieorientowanego? Opisz struktury danych jakich używa algorytm znajdowania spójnych składowych. Opisz algorytm znajdowania spójnych składowych grafu nieorientowanego. Jaka jest złożoność tego algorytmu? Jak będzie wyglądał efekt wykonania algorytmu znajdowania spójnych składowych grafu nieorientowanego na grafie G reprezentowanym przez następujące listy incydencji:

```
1 : 3 5
2 : 3 4
3 : 1 2
4 : 2
5 : 1 8
6 : 7
7 : 6
8 : 5
```

**Egzamin ze Wstępu do Informatyki.
12 czerwca 2012.**

Zadanie 1

Dana jest tablica $P[1..n]$ liczb całkowitych obliczona przez procedurę BFS w czasie przeszukiwania wszerz grafu $G = (V, E)$ z wierzchołka s taka, że

$$P[v] = \begin{cases} 0 & \text{gdy } v = s \text{ lub nie ma drogi z } s \text{ do } v \\ w & \text{jeśli } w \text{ jest przedostatnim wierzchołkiem na najkrótszej ścieżce z } s \text{ do } v \end{cases}$$

dla $v \in V = \{1, \dots, n\}$. Napisać funkcję, która dla danego wierzchołka v tworzy listę jednokierunkową wierzchołków najkrótszej ścieżki z s do v . Jeśli nie ma ścieżki z s do v funkcja ma zwracać głowę do listy pustej.

Zadanie 2

Napisać funkcję, która dla danego drzewa binarnego zwraca korzeń poddrzewa o maksymalnej liczbie wierzchołków, którego każdy wierzchołek ma parzystą liczbę synów (0 lub 2).

Zadanie 3

Dany jest graf niezorientowany $G = (V, E)$ i podzbiór jego wierzchołków $X \subseteq V$. Opisać algorytm, który oblicza odległość wierzchołków od zbioru X . Odległość wierzchołka v od zbioru X w grafie G to minimum z długości najkrótszych dróg z wierzchołków X do v .

Zadanie 4

Co to jest sortowanie topologiczne grafu skierowanego? Opisz algorytm sortujący topologicznie graf skierowany. Jakie grafy skierowane można posortować topologicznie? Jaka jest złożoność algorytmu sortowania topologicznego? Jak będzie wyglądał efekt wykonania algorytmu sortowania topologicznego na grafie G reprezentowanym przez następujące listy incydencji:

```
1 : 3 5 6
2 : 3
3 :
4 : 2
5 :
6 :
7 : 6
8 :
```

**Egzamin Poprawkowy ze Wstępu do Informatyki.
8 września 2011.**

Zadanie 1

Dane są dwie listy: pierwsza zawiera liczby ujemne i jest posortowana rosnąco, druga zawiera liczby dodatnie i jest posortowana malejąco. Napisać procedurę, która połączy te listy w jedną posortowaną rosnąco.

Zadanie 2

Drzewo BST o kluczach całkowitych jest k -rzadkie, jeśli moduł różnicy kluczy z każdych dwóch różnych wierzchołków jest większy od k . Napisz procedurę, która dla danego drzewa BST o kluczach całkowitych i liczby k sprawdzi czy to drzewo jest k -rzadkie.

Zadanie 3

Dla danego grafu skierowanego $G = (V, E)$ i zbioru wierzchołków $X \subseteq V$ sprawdzić czy z każdego wierzchołka grafu można dojść do pewnego wierzchołka w zbiorze X .

UWAGA. To zadanie można rozwiązać opisując algorytm słownie (tzn. nie jako procedurę w Języku Pascal) używając do opisu algorytmów poznanych na wykładzie.

Zadanie 4

Zdefiniuj co to jest minimalne drzewo rozpinające grafu z wagami G . Jaka jest złożoność algorytmu znajdowania minimalnego drzewa rozpinającego? Jak by się zmieniła złożoność tego algorytmu, gdyby graf reprezentować przy pomocy macierzy incydencji?

Naszkicuj minimalne drzewo rozpinające dla grafu G z funkcją wag w reprezentowanych przez następujące listy incydencji:

1 :	(2, 2)	(4, 6))		
2 :	(1, 2)	(3, 4)	(6, 3)	(5, 2)
3 :	(2, 4)	(6, 2)		
4 :	(1, 6)	(5, 4)	(8, 6)	
5 :	(2, 2)	(7, 1)	(4, 4)	(9, 3)
6 :	(3, 2)	(2, 3)	(7, 6)	
7 :	(6, 6)	(5, 1)	(9, 7)	
8 :	(4, 6)	(9, 4)		
9 :	(5, 3)	(7, 7)	(8, 4)	

Para na liście oznacza koniec krawędzi i jej wagę. Na przykład druga para na pierwszej liście (4,6) oznacza, że krawędź z 1 do 4 ma wagę 6.

Egzamin ze Wstępu do Informatyki. 6 czerwca 2011.

Zadanie 1

Dane są 3 listy jednokierunkowe liczb całkowitych posortowane rosnąco. Napisać procedurę, która znajduje wszystkie liczby, które należą do trzech list na raz i usuwa je ze wszystkich trzech.

Zadanie 2

Napisać funkcję, która dla danego drzewa BST o kluczach całkowitych oblicza, ile jest różnych kluczy w tym drzewie.

Zadanie 3

Dla danego grafu skierowanego G , znaleźć moc minimalnego zbioru wierzchołków X takiego, że do każdego wierzchołka w grafie G można dojść z co najmniej jednego wierzchołka ze zbioru X .

UWAGA. To zadanie można rozwiązać opisując algorytm słownie (tzn. nie jako procedurę w Języku Pascal) używając do opisu algorytmów poznanych na wykładzie.

Zadanie 4

Zdefiniuj co to jest drzewo przeszukiwania wszerz grafu G . Jaka jest złożoność algorytmu przeszukiwania wszerz grafu G ? Jak by się zmieniła złożoność algorytmu BFS, gdyby graf reprezentować przy pomocy macierzy incydencji?

Naszkicuj drzewo przeszukiwania wszerz grafu G z wierzchołka $s = 3$ gdy graf G jest reprezentowany przez następujące listy incydencji:

```

1 : 2 5
2 : 6 5
3 : 7 2 4
4 : 11 8
5 :
6 : 5
7 : 11 9
8 :
9 :
10 : 6
11 : 10 12
12 : 8

```

Egzamin Poprawkowy ze Wstępu do Informatyki. 9 września 2010.

Każde z poniższych zadań ma rozwiązanie liniowe w stosunku do rozmiaru danych wejściowych. Rozwiązania o złożoności większej będą oceniane w skali od 0 do 8 punktów. Każde zadanie należy rozwiązać na osobnej kartce.

1. Dana jest lista cykliczna liczb całkowitych taka, że suma elementów tej listy jest nieujemna. Napisać procedurę, która przestawi tak głowę listy (nie przestawiając elementów na liście) by wszystkie sumy częściowe liczone począwszy od głowy były nieujemne. Na przykład, po wykonaniu procedury dla listy cyklicznej (2,-1,-2,10,-5,-7,3,0) głowa powinna wskazywać element 3.
2. Napisać procedurę, która dla danego drzewa binarnych poszukiwań (BST) zwraca wskaźnik do korzenia maksymalnego poddrzewa w którym każdy klucz występuje co najwyżej jeden raz.
3. Dla danego grafu skierowanego G (reprezentowanego przez listy incydencji), wierzchołek v i liczby naturalnej k , obliczyć liczbę wierzchołków u w tym grafie takich, że łączna długość dróg z v do u i z u do v jest nie większa niż k .

Uwaga. To zadanie można rozwiązać pisząc procedurę w języku Pascal lub też opisując słowami algorytm używając procedur opisanych na wykładzie.

Egzamin ze Wstępu do Informatyki. 7 czerwca 2010.

Każde z poniższych zadań ma rozwiązanie liniowe w stosunku do rozmiaru danych wejściowych. Rozwiązania o złożoności większej będą oceniane w skali od 0 do 8 punktów.

1. Mówimy, że ciąg liczbowy $a_1, \dots, a_N$ oscyluje jeśli dla każdego n z przedziału $2 \leq n \leq N-1$ spełniony jest warunek: $(a_{n-1} - a_n) * (a_n - a_{n+1}) \leq 0$.
Dana jest lista (jednokierunkowa) liczb całkowitych. Napisać procedurę, która zamieni kolejność elementów na tej liście w taki sposób, by reprezentowany przez nią ciąg oscylował.
Np. dla listy: 4, 12, 18, 6, 0, 29, 75, 8, 72, 36, 56, 59, 82, 73, 23, 85, 83, 55, 83, 62 jednym z poprawnych wyników będzie lista: 4, 18, 6, 12, 0, 75, 8, 72, 29, 56, 36, 82, 59, 73, 23, 85, 55, 83, 83, 62
2. Napisać procedurę, która dla danego drzewa binarnego (niekoniecznie drzewa binarnych poszukiwań (BST)!) zwraca wskaźnik do korzenia maksymalnego poddrzewa BST tego drzewa binarnego.

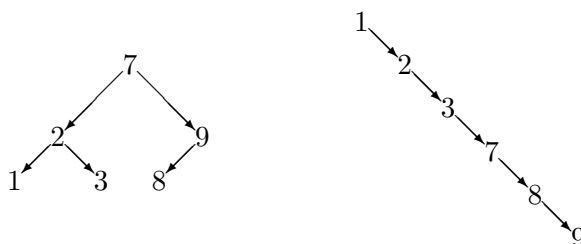
3. Niech $G = (V, E)$ będzie grafem zorientowanym, v wierzchołkiem w V oraz X podzbiorem niepustym zbioru V . *Odległość* od zbioru X do v to minimum z długości dróg z elementów X do wierzchołka v . Napisać funkcję, która dla danego grafu $G = (V, E)$ reprezentowanego przez listy incydencji, wierzchołka v i niepustej listy wierzchołków X oblicza odległość od X do v .

Egzamin ze Wstępu do Informatyki. 31 sierpnia 2009.

1. Napisz funkcję, która dla danego grafu skierowanego G reprezentowanego przez listy incydencji, wierzchołka x , i liczby naturalnej dodatniej k , obliczy ile jest wierzchołków w grafie G o odległości k od x takich, z których można z powrotem dojść do x .
2. Dana jest lista liczb całkowitych. Napisać procedurę, która rozdzieli tę listę na 3 listy z zachowaniem uporządkowania, w taki sposób, żeby jedna lista zawierała elementy podzielne przez 2, druga podzielne przez 3 ale nie przez 2, a trzecia pozostałe elementy.
3. Dane jest drzewo, którego węzły zawierają liczby całkowite. Napisać funkcję, która zwraca wskaźnik do korzenia maksymalnego pod względem wysokości poddrzewa zawierającego same liczby parzyste.

Egzamin ze Wstępu do Informatyki. 8 czerwca 2009.

1. Napisz procedurę, która z elementów drzewa BST o kluczach całkowitych tworzy uporządkowaną niemalejąco listę, w taki sposób, żeby dowiązaniem do następnego elementu listy było pole 'prawe'. Na przykład z drzewa po lewej stronie procedura powinna utworzyć listę-drzewo po prawej stronie.



2. Napisz procedurę, która przerabia dowolne drzewo BST na drzewo BST o tych samych elementach ale takie, w którym wierzchołki mają co najwyżej prawych synów.
3. Napisz funkcję sprawdzającą czy w grafie skierowanym $G = (V, E)$ reprezentowanym jako listy incydencji istnieje taki wierzchołek, z którego można dojść do każdego innego.
Rozwiązania o złożoności powyżej $O(|V| + |E|)$ będą punktowane w skali od 0 do 4 punktów.
4. Napisz funkcję, która dla danego drzewa BST zwraca liczbę takich liści, do których droga od korzenia biegnie po większej liczbie lewych synów niż prawych.

Egzamin ze Wstępu do Informatyki. 1 września 2008.

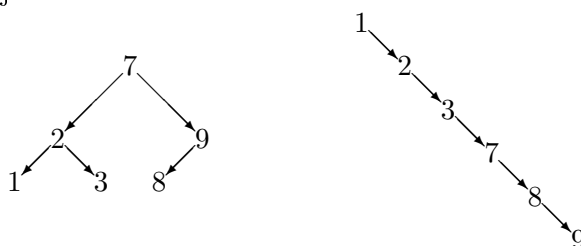
1. Napisz procedurę usuwającą z danej listy uporządkowanej jednokierunkowej liczb całkowitych wszystkie elementy powtarzające się.
2. Napisz funkcję, która dla danego drzewa binarnego, zwraca wskaźnik do najwyższego poddrzewa, które ma jednakową liczbę węzłów w obu swoich poddrzewach.

3. Napisz funkcję, która dla grafu skierowanego $G = (V, E)$ reprezentowanego przez listy incydencji zwraca liczbę wierzchołków z których można dojść do każdego wierzchołka tego grafu.

Rozwiązanie, oprócz procedury lub procedur, powinno zawierać krótki opis słowny implementowanego algorytmu. Rozwiązania o złożoności powyżej $O(|V| + |E|)$ będą punktowane w skali od 0 do 4 punktów.

Egzamin ze Wstępu do Informatyki. 12 czerwca 2008.

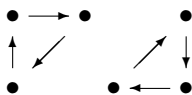
1. Napisz funkcję, która dla danego drzewa BST zwraca liczbę takich liści, do których droga od korzenia biegnie po większej liczbie lewych synów niż prawych.
2. Napisz procedurę, która z elementów drzewa BST o kluczach całkowitych tworzy uporządkowaną niemalejąco listę, w taki sposób, żeby dowiązaniem do następnego elementu listy było pole 'prawe'. Na przykład z drzewa po lewej stronie procedura powinna utworzyć listę-drzewo po prawej stronie.



3. Napisz funkcję sprawdzającą czy w grafie skierowanym $G = (V, E)$ reprezentowanym jako listy incydencji istnieje taki wierzchołek, z którego można dojść do każdego innego. Rozwiązania o złożoności powyżej $O(|V| + |E|)$ będą punktowane w skali od 0 do 4 punktów.

Egzamin ze Wstępu do Informatyki. 6 września 2007.

1. *Grubość* drzewa binarnego jest najmniejsza liczba k taka, że każda ścieżka w drzewie ma nie więcej niż k strzałek w lewo i nie więcej niż k strzałek w prawo. Napisać funkcję obliczającą grubość danego drzewa binarnego.
2. Napisać procedurę, która dla danego drzewa binarnych poszukiwań o kluczach całkowitych zwraca wskaźnik do korzenia poddrzewa tego drzewa o maksymalnej liczbie węzłów, w którym wszystkie klucze są parzyste.
3. Dany jest graf skierowany reprezentowany jako listy incydencji. Sprawdzić czy ten graf jest sumą rozłączną dwóch cykli prostych o równej długości. Na przykład poniższy graf jest sumą dwóch cykli rozłącznych o długości 3:



Egzamin ze Wstępu do Informatyki. 14 czerwca 2007.

1. Lista jednokierunkowa jest okresowa, jeśli można ją podzielić na k ($k \geq 2$) takich samych kawałków. Napisać funkcję, która sprawdza czy dana lista jednokierunkowa jest okresowa. Na przykład lista 5 2 7 8 5 2 7 8 jest okresowa, a 5 2 7 8 5 2 nie jest.

2. Napisać procedurę, która dla danego drzewa binarnych poszukiwań o kluczach całkowitych wypisuje w porządku rosnącym te liczby całkowite pomiędzy kluczem najmniejszym i największym, które nie występują jako klucze w tym drzewie.
3. Dany jest graf skierowany reprezentowany jako listy incydencji. Napisać procedurę, która usuwa z tego grafu te krawędzie, których końce leżą w różnych silnie spójnych składowych tego grafu.

Egzamin ze Wstępu do Informatyki. 7 września 2006.

Zadanie 1

Dane są 2 listy: l1 posortowana względem nazwisk, element zawiera nazwisko i adres, l2 posortowana względem nazwisk, element zawiera nazwisko i rok urodzenia. Napisać procedurę tworzącą listę l3, która składa się z elementów zawierających nazwisko, adres i rok urodzenia pobrane z l1 i l2. Informacje użyte do utworzenia l3 usunąć z l1 i l2. Jeśli w jednej z list l1 lub l2 brak nazwiska, to element ten pozostawić i nie tworzyć odpowiedniego elementu w l3. Tzn. po zakończeniu procedury l1 i l2 mają zawierać tylko te elementy, dla których nie było odpowiednika w drugiej liście.

Uwaga. Wszystkie typy dotyczące list należy zdefiniować.

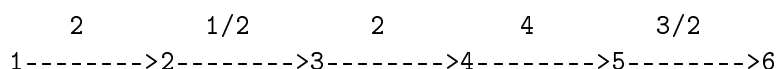
Zadanie 2

Napisać funkcję, która dla danego drzewa binarnego zwraca wskaźnik do jego liścia będącego końcem gałęzi, na której suma kluczy jest maksymalna.

Zadanie 3

Napisać funkcję, która dla danego grafu zorientowanego $G = (V, E)$, ($V = \{1, \dots, n\}$) z nieujemną funkcją wag $w : E \rightarrow R$ i wierzchołków $s, t \in V$ zwraca numer wierzchołka $m \in V$, który jest najbliżej środka jednej z najkrótszych ścieżek z s do t .

Przykład. Dla grafu



$s=1$ oraz $t=6$ odpowiedź powinna wynosić $m=4$.

Wskazówka. Użyj algorytmu Dijkstry. Samej procedury `procedure Dijkstra(G,w,s)`, o ile jest to procedura z wykładu nie trzeba implementować, wystarczy opisać jej efekt i się do niej odwołać w stosownym momencie.

Egzamin ze Wstępu do Informatyki. 27 czerwca 2006.

Zadanie 1

Napisać procedurę, która drukuje liczby całkowite z listy jednokierunkowej liczb całkowitych uporządkowanej niemalejąco w porządku nierosnącym.

Zadanie 2

Drzewo binarnych poszukiwań jest *zbalansowane* jeśli dla każdego wierzchołka tego drzewa różnica wielkości jego poddrzew lewego i prawego jest co najwyżej 1.

Napisać funkcję, która dla danego drzewa binarnych poszukiwań zwraca wskaźnik do wierzchołka, który jest korzeniem poddrzewa zbalansowanego o maksymalnej wysokości.

Zadanie 3

Napisać funkcję, która dla danego grafu zorientowanego $G = (V, E)$ z nieujemną funkcją wag

$w : E \rightarrow R$ i wierzchołka $s \in V$ zwraca ciężar najlżejszego cyklu przechodzącego przez s . Ciężarem cyklu nazywamy sumę wag krawędzi tego cyklu.

Wskazówka. Użyj algorytmu Dijkstry. Samej procedury `procedure Dijkstra(G,w,s)`, o ile jest to procedura z wykładu nie trzeba implementować, wystarczy opisać jej efekt i się do niej odwołać w stosownym momencie.

Każde zadanie należy rozwiązać na osobnej kartce.

Egzamin ze Wstępu do Informatyki. 1 września 2005

Zadanie 1

Lista jednokierunkowa zbudowana jest z elementów zadeklarowanych jako:

```
type
  wsk=^elem;
  elem =record
    id: integer;
    nast: wsk
  end;
```

Dane są dwie listy l1 i l2 uporządkowane rosnąco. Utworzyć listę uporządkowaną rosnąco powstałą po usunięciu z listy l1 elementów występujących na liście l2 (identyfikowanych przez pole id). Wynik zwrócić jako l1, a l2 powinna pozostać bez zmian.

Zadanie 2

Drzewo binarne jest *zrównoważone*, jeśli dla każdego węzła spełniony jest warunek:

(i) liczba węzłów w prawym poddrzewie=liczba węzłów w lewym poddrzewie.

Drzewo binarne jest zdefiniowane następująco:

```
drzewo = ^wezel; wezel = record
  klucz: integer;
  lewy, prawy: drzewo
end;
```

Napisać funkcję *function zrownowazone(d:drzewo):boolean* zwracającą wartość **true**, jeśli drzewo jest zrównoważone, a **false** w przeciwnym przypadku.

Zadanie 3

Krawędź w spójnym grafie niezorientowanym nazywamy *mostem*, jeżeli po jej usunięciu graf rozpada się na dwie składowe. Napisać procedurę, która dla danego grafu spójnego (reprezentowanego w postaci list incydencji) oraz danej jego krawędzi (reprezentowanej w postaci pary liczb całkowitych - węzłów grafu) sprawdza czy jest ona mostem.

Egzamin ze Wstępu do Informatyki. czerwiec 2005.

Zadanie 1

Lista jednokierunkowa zbudowana jest z elementów zadeklarowanych jako:

```
type
  wsk=^elem;
  elem =record
    id: integer;
    nast: wsk
  end;
```

Dane są dwie listy l1 i l2 uporządkowane rosnąco. Utworzyć listę uporządkowaną rosnąco powstałą po usunięciu z listy l1 elementów występujących na liście l2 (identyfikowanych przez pole id). Wynik zwrócić jako l1, a l2 powinna pozostać bez zmian.

Zadanie 2

Napisać procedurę, która wywołana od korzenia drzewa BST zwraca liczbę węzłów w tym drzewie o parzystej wysokości.

Zadanie 3

Krawędź w spójnym grafie nieorientowanym nazywamy *mostem* o ile po jej usunięciu graf rozpada się na dwie składowe. Napisać procedurę, która dla danego grafu spójnego G reprezentowanego jako przez listy incydencji i jego krawędzi e sprawdza czy e jest mostem w G .

Egzamin ze Wstępu do Informatyki. 31 Maja 2004.

Część II: Zadania.

Zadanie 1

Zbiór wierzchołków $X \subseteq V$ grafu $G = (V, E)$ jest *klanem* jeśli dla każdego $a \in V - X$ i każdych $x, y \in X$, krawędź (x, a) należy do E wtedy i tylko wtedy gdy krawędź (y, a) należy do E .

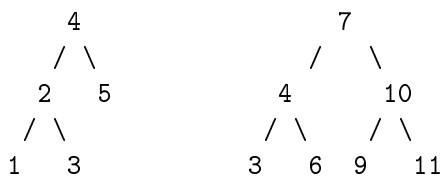
Dla danego grafu $G = (V, E)$ ($V = \{1, \dots, n\}$), reprezentowanego przez listy incydencji uporządkowane niemalejąco, i liczb $1 \leq k < l \leq n$ sprawdzić czy zbiór $\{k, l\}$ jest klanem.

Zadanie 2

Drzewo binarnych poszukiwań o kluczach całkowitych jest *gęste* jeśli każdy klucz pomiędzy minimalnym a maksymalnym występuje dokładnie jeden raz.

Napisać procedurę, która, dla danego drzewa binarnych poszukiwań, zwraca klucz do korzenia gęstego poddrzewa o maksymalnej liczbie wierzchołków.

Na przykład spośród drzew



lewe jest gęste a prawe nie. W pierwszym przypadku procedura powinna zwrócić wskaźnik do korzenia, wierzchołka o kluczu 4, a w drugim procedura powinna zwrócić wskaźnik do wierzchołka o kluczu 10.

Egzamin ze Wstępu do Informatyki. 27 maja 2003.

Zadanie 1

Waga drzewa BST o kluczach rzeczywistych jest to suma kluczy wszystkich jego wierzchołków. Napisać funkcję w języku Pascal, która dla danego drzewa BST o kluczach rzeczywistych zwraca wskaźnik do wierzchołka poddrzewa, którego waga jest najbliższa 0.

Zadanie 2

Napisać procedurę w języku Pascal, która przedstawia tak elementy jednokierunkowej listy liczb całkowitych by nieujemne liczby stały przed ujemnymi. (Rozwiązania o złożoności większej niż $O(n)$ będą oceniane w skali 0-4 punkty.)

Zadanie 3

Drzewo binarne jest zdefiniowane następująco:

```
type drzewo = ^wezel;  
wezel = record  
    klucz: integer;  
    lewy, prawy: drzewo  
end;
```

Napisz funkcję *rowne*(*d1*, *d2*: *drzewo*):*boolean*, która zwraca *true*, jeśli drzewa *d1* i *d2* są jednakowe (taka sama struktura i te same wartości w odpowiednich węzłach), a *false* w przeciwnym przypadku.

Zadanie 4

Ścieżkę w grafie nazywamy prostą, o ile wszystkie wierzchołki na niej występujące, z wyjątkiem być może pierwszego i ostatniego, są różne. Napisać funkcję w języku Pascal, która dla danej pary wierzchołków *k*, *l* grafu skierowanego *G* o *n* wierzchołkach, reprezentowanego jako listy incydencji, zwraca liczbę ścieżek prostych pomiędzy tymi wierzchołkami.

Egzamin ze Wstępu do Informatyki. 10 czerwca 2002

Zadanie 1

Napisać funkcję w języku Pascal, która sprawdza czy dany graf skierowany reprezentowany przez listy incydencji jest dwudzielny.

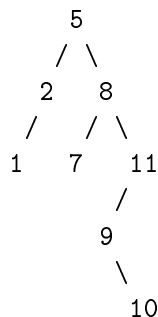
Graf skierowany $G = (V, E)$ jest grafem dwudzielnym, jeśli istnieje podział zbioru V na zbiory V_1 i V_2 takie, że dla dowolnej krawędzi $(i, j) \in E$ mamy $i \in V_1$ oraz $j \in V_2$.

Zadanie 2

Napisać procedurę w języku Pascal, która z listy jednokierunkowej liczb całkowitych usuwa elementy mniejsze od 0.

Zadanie 3

Napisać funkcję w języku Pascal, która dla danego drzewa binarnych poszukiwań zwraca wskaźnik do wierzchołka o maksymalnej wysokości, który ma dokładnie jeden następnik. Np. dla drzewa



funkcja powinna zwrócić wskaźnik do węzła zawierającego klucz 11.

Egzamin ze Wstępu do Informatyki. 5 września 2000

Zadanie 1

W programie zadeklarowana jest stała N oraz typ `type tab:array[1..N] of char`. Napisać w języku Pascal procedurę: *procedure shift*(*var A:tab*; *m:integer*) realizującą przesunięcie cykliczne elementów tablicy *A* o *m* pozycji w lewo. Np. dla $A = [a, b, c, d, e]$ wynik działania *shift*(*A*, 2) powinien być $A = [c, d, e, a, b]$.

Zadanie 2

Węzeł drzewa binarnego zdefiniowany jest deklaracją:

```
type
  wsk=^elem;
  elem=record
    id:integer;
    ldow,pdow:wsk
  end;
```

Napisać w języku Pascal funkcję *function lw(t:wsk):integer*, której wartością jest liczba węzłów drzewa binarnego o korzeniu wskazanym przez zmienną wskaźnikową *t*, w których pole *klucz* ma tę samą wartość co pole *klucz* korzenia.

Zadanie 3

Lista jednokierunkowa zbudowana jest z elementów zadeklarowanych jako:

```
type
  wsk=^elem;
  elem=record
    id:integer;
    nast:drzewo
  end;
```

Lista *l* jest listą dowolnych liczb całkowitych. Napisać w języku Pascal funkcję *rozdziel(var l,lu,ld:wsk):wsk*, która z listy *l* usunie elementy zawierające w polu *id* liczby ujemne i utworzy z nich listę *lu* oraz elementy zawierające w polu *id* liczby dodatnie i utworzy z nich listę *ld*. Ewentualne zera zostaną na liście *l*.

Egzamin ze Wstępu do Informatyki. maj 2000

Zadanie 1

Dana jest tablica $A[1..n, 1..m]$ of *integer* posortowana rosnąco i wierszami i kolumnami oraz liczba *x*. Napisać program sprawdzający czy *x* występuje w *A*. Należy zwrócić uwagę na efektywność programu.

Preferowane są rozwiązania działające w czasie $O(n + m)$. Rozwiązania rzędu $O(n * m)$ będą punktowane w skali 0-4pkt.

Zadanie 2

Wektor rzadki $\vec{x} = (x_1, \dots, x_n)$ reprezentowany jest jako lista niezerowych elementów x_i . Element tej listy jest zadeklarowany jako:

```
type
  wsk=^elem;
  elem=record
    wart:integer;
    ind:wsk;
    nast:wsk
  end;
```

wart oznacza wartość x_{ind} , a *nast* dowiązanie do następnego niezerowego elementu. Lista jest uporządkowana według rosnącej kolejności indeksów.

Napisać w języku Pascal funkcję *function ilskal(x,y:wsk):real*, której wartością jest iloczyn skalarny wektorów rzadkich *x* i *y*.

Zadanie 3

Drzewo binarne jest zdefiniowane następująco:

```
type
  drzewo = ^wezel;
  wezel = record
    klucz: integer;
    lewy, prawy: drzewo
  end;
```

Napisać w języku Pascal funkcję *rowne*(*var d1, d2: drzewo*):*boolean*, która zwraca **true**, jeśli drzewa *d1* i *d2* są jednakowe (taka sama struktura i te same wartości w odpowiednich węzłach), a **false** w przeciwnym przypadku.

Egzamin ze Wstępu do Informatyki. 6 września 1999

Zadanie 1

Na wejściu dany jest plik tekstowy *tekst.txt*. Napisać program w języku Pascal, który znajduje linie o maksymalnej liczbie słów i zwraca numer tej linii. Przez słowo rozumiemy dowolny ciąg znaków różnych od spacji, końca linii i końca pliku. Linie numerowane są od 1.

Zadanie 2

Lista jednokierunkowa zbudowana jest z elementów zadeklarowanych jako:

```
type
  wsk = ^elem;
  elem = record
    id: integer;
    nast: wsk
  end;
```

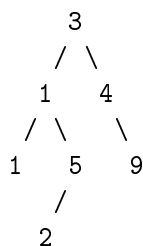
Lista *l* jest liczbą dowolnych liczb całkowitych. Napisać w języku Pascal funkcję *rozdziel*(*var l, lu, ld: wsk*):*wsk*, która z listy *l* usunie elementy zawierające w polu *id* liczby ujemne i utworzy z nich listę *lu* oraz elementy zawierające w polu *id* liczby dodatnie i utworzy z nich listę *ld*. Ewentualne zera zostaną na liście *l*.

Zadanie 3

Drzewo binarne jest zdefiniowane następująco:

```
type
  drzewo = ^wezel;
  wezel = record
    klucz: integer;
    lewy, prawy: drzewo
  end;
```

Napisać w języku Pascal funkcję *ściezka*(*d: drzewo*):*integer*, która znajduje w drzewie *d* ścieżkę o maksymalnej sumie kluczy i zwraca jako wartość tę sumę. Np. dla drzewa



funkcja powinna zwrócić wartość 16.

Egzamin ze Wstępu do Informatyki. 24 maja 1999

Zadanie 1

Na wejściu dany jest plik tekstowy *tekst.txt*. Napisać program w Pascalu, który liczy ile słów występuje w tym pliku. Przez słowo rozumiemy dowolny ciąg znaków różnych od spacji, końca linii i końca pliku.

Zadanie 2

Lista jednokierunkowa zbudowana jest z elementów zadeklarowanych jako:

```
type
  wsk=^elem;
  elem=record
    id:integer;
    nast:wsk
  end;
```

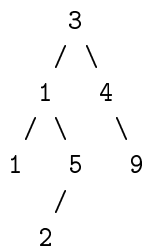
Napisać w języku Pascal funkcję *merge(var l1,l2:wsk):wsk*. Listy *l1* i *l2* są posortowane rosnąco względem pola *id*. Wartością funkcji *merge* jest lista posortowana rosnąco zbudowana z elementów list *l1*, *l2* (listy te powinny stać się puste po wykonaniu *merge*).

Zadanie 3

Drzewo binarne jest zdefiniowane następująco:

```
type
  drzewo=^wezel;
  wezel=record
    klucz:integer;
    lewy,prawy:drzewo
  end;
```

Napisać w języku Pascal funkcję *ścieżka(d: drzewo):integer*, która oblicza sumę kluczy na ścieżce o maksymalnej długości w tym drzewie; np. dla drzewa



funkcja powinna zwrócić wartość 9.

Egzamin ze Wstępu do Informatyki. 21 maja 1998

Zadanie 1

Węzeł drzewa binarnego zdefiniowany jest deklaracją

```

type
  wsk=^elem;
  elem=record
    klucz:integer;
    ldow, pdow:wsk;
  end;

```

Napisać w języku Pascal funkcję: *function bst(t:wsk):boolean*, której wartością jest **true**, jeśli *t* jest binarnym drzewem poszukiwań, a **false** w przeciwnym przypadku.

Zadanie 2

Węzeł drzewa binarnego zdefiniowany jest jak w zadaniu 1. Napisać w języku Pascal procedurę: *procedure dołącz(t1,t2:wsk)*, gdzie *t1* jest binarnym drzewem poszukiwań, a *t2* jest dowolnym drzewem binarnym. W wyniku wykonania *dołącz t1* jest korzeniem drzewa binarnych poszukiwań uzyskanym w wyniku dołączenia elementów drzewa *t2* do drzewa *t1*.

Zadanie 3

Lista jednokierunkowa zbudowana jest z elementów zadeklarowanych jako:

```

type
  wsk=^elem;
  elem=record
    id:integer;
    nastwsk
  end;

```

napisać funkcję: *function znajdz(l1,l2,l3:wsk):integer*. Listy *l1*, *l2*, *l3* są listami osób, których unikalne identyfikatory są umieszczone w polu *id*. Listy są posortowane rosnąco względem pola *id*. Wartością funkcji *znajdz* jest identyfikator osoby, która występuje na wszystkich trzech listach lub 0, jeśli takiej osoby nie ma.

Zadanie 4

Typ graf jest tablicą list jednokierunkowych.

```

type
  wsk=^elem;
  elem=record
    id:integer;
    nast:wsk
  end;
  graf=array[1..N] of wsk;

```

N jest stałą zadeklarowaną w programie głównym. Napisać funkcję: *function rl(A:graf):boolean*. *A[i]* jest listą węzłów, do których wychodzi strzałka z węzła *i*. Wartością funkcji jest **true**, jeśli zbiór węzłów $\{1, \dots, N\}$ można podzielić na dwa rozłączne podzbiory P_1 i P_2 takie, że

1. jeśli $x \in P_1$, to z x wychodzi strzałka co najwyżej do węzła $y \in P_2$;
2. jeśli $x \in P_2$, to z x nie wychodzi żadna strzałka.

W przeciwnym przypadku wartością funkcji jest **false**.

Listy

Zadanie 1

Napisać procedurę w języku Pascal, która wstawia liczbę całkowitą na listę jednokierunkową liczb całkowitych uporządkowaną niemalejąco.

Zadanie 2

Napisać procedurę w języku Pascal porządkującą listę jednokierunkową.

Zadanie 3

Napisać procedurę w języku Pascal niszczącą listę jednokierunkową.

Zadanie 4

Dla danej liczby naturalnej n stworzyć listę jednokierunkową dzielników n .

Zadanie 5

Napisać procedurę, która łączy dwie listy jednokierunkowe.

Zadanie 6

Napisać procedurę, która łączy dwie listy jednokierunkowe uporządkowane niemalejąco w jedną listę jednokierunkową uporządkowaną niemalejąco.

Zadanie 7

Napisać procedurę, która dla danego pliku tekstowego zwraca listę słów występujących w tym pliku wraz z częstością ich wystąpień.

Zadanie 8

Używając list wydrukować w odwrotnej kolejności plik liczb całkowitych .

Zadanie 9

Lista jednokierunkowa zbudowana jest z elementów zadeklarowanych jako:

```
type
  wsk=^elem;
  elem =record
    id: integer;
    nast: wsk
  end;
```

Dane są dwie listy l1 i l2 uporządkowane rosnąco. Utworzyć listę uporządkowaną rosnąco powstałą po usunięciu z listy l1 elementów występujących na liście l2 (identyfikowanych przez pole id). Wynik zwrócić jako l1, a l2 powinna pozostać bez zmian.

Zadanie 10

Dany jest ciąg liczb rzeczywistych . Rozbić ten ciąg na dwa podciągi niemalejące: $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ oraz $a_{j_1}, a_{j_2}, \dots, a_{j_l}$ takie, że: $l + k = n$, $i_1 < i_2 < \dots < i_k$, $j_1 < j_2 < \dots < j_l$ oraz $\{i_1 \dots i_k\} \cup \{j_1 \dots j_l\} = \{1, \dots, n\}$.

Uwaga. Warunkiem koniecznym i wystarczającym na to by istniało takie rozbiecie ciągu $a_1, a_2, \dots, a_n$ jest to by w tym ciągu nie istniał podciąg malejący długości 3, tzn. by nie istniały trzy liczby $1 \leq s_1 < s_2 < s_3 \leq n$ takie, że $a_{s_1} > a_{s_2} > a_{s_3}$.

Zadanie 11

Dana jest lista jednokierunkowa, której elementami są liczby rzeczywiste (napisz stosowną definicję typu). Napisz funkcję, której wartością jest lista w odwróconej kolejności (np. dla listy 3,1,4,1,5,9 funkcja powinna zwrócić listę 9,5,1,4,1,3).

Zadanie 12

Dana jest lista jednokierunkowa, której elementami są liczby rzeczywiste (napisz stosowną definicję typu). Napisz funkcję, której wartością jest lista w odwróconej kolejności (np. dla listy 3,1,4,1,5,9 funkcja powinna zwrócić listę 9,5,1,4,1,3).

Zadanie 13

Wektor rzadki $\vec{x} = (x_1, \dots, x_n)$ reprezentowany jest jako lista niezerowych elementów x_i . Element tej listy jest zadeklarowany jako:

```
type
  wsk=^elem;
  elem=record
    wart:integer;
    ind:wsk;
    nast:wsk
  end;
```

wart oznacza wartość x_{ind} , a **nast** dowiązanie do następnego niezerowego elementu. Lista jest uporządkowana według rosnącej kolejności indeksów.

Napisać w języku Pascal funkcję *function suma(x,y:wsk):real*, której wartością jest suma wektorów rzadkich x i y .

Zadanie 14

Napisać procedurę w języku Pascal, która usuwa z listy jednokierunkowej liczb całkowitych liczby ujemne.

Drzewa binarnych poszukiwań

Zadanie 15

Napisać procedurę w języku Pascal obliczającą:

1. liczba wierzchołków drzewa BST;
2. liczba liści drzewa BST;
3. wysokość drzewa BST;
4. liczbę wierzchołków o wysokości k ;
5. liczbę wierzchołków o głębokości k .

Zadanie 16

Napisać procedurę w języku Pascal sprawdzającą czy dwa drzewa BST mają te same klucze (to znaczy każdy klucz występuje tą samą liczbę razy w obu drzewach).

Zadanie 17

Napisać procedurę w języku Pascal sprawdzającą czy jedno drzewo BST jest poddrzewem drugiego.

Zadanie 18

Napisać procedurę w języku Pascal obliczającą wskaźnik do najdłuższej gałęzi w drzewie BST.

Zadanie 19

Napisać procedurę w języku Pascal obliczającą wskaźnik do największego pełnego poddrzewa binarnego w drzewie BST.

Zadanie 20

Napisać procedurę w języku Pascal sprawdzającą czy etykietowane drzewo binarne jest drzewem BST.

Zadanie 21

Napisać procedurę w języku Pascal sprawdzającą czy drzewo BST jest zbalansowane. Drzewo jest *zbalansowane* o ile różnica wielkości poddrzew każdego wierzchołka jest co najwyżej 1.

Zadanie 22

Napisać procedurę w języku Pascal obliczającą wskaźnik do wierzchołka o najmniejszej różnicy z daną liczbą rzeczywistą (zakładamy, że klucze w drzewie BST są rzeczywiste).

Zadanie 23

Napisać procedurę w języku Pascal, która czyści drzewo BST.

Zadanie 24

Drzewo binarne jest zdefiniowane następująco:

```
drzewo = ^wezel; wezel = record
    klucz: integer;
    lewy, prawy: drzewo
end;
```

Napisać funkcję *function ile(d:drzewo):integer*, która zwraca liczbę węzłów w drzewie *d* o kluczach mniejszych od klucza korzenia drzewa.

Zadanie 25

Napisać procedurę, która wywołana od korzenia drzewa BST zwraca liczbę węzłów w tym drzewie o parzystej wysokości.

Zadanie 26

Napisać procedurę w języku Pascal, która dla danego drzewa BST o kluczach całkowitych, znajduje wskaźnik do korzenia maksymalnego poddrzewa, o kluczach parzystych.

Zadanie 27

Napisać procedurę w języku Pascal, która sprawdza czy drzewo BST o podanym korzeniu jest pełnym drzewem binarnym.

Zadanie 28

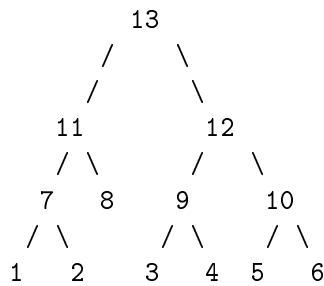
Napisać procedurę w języku Pascal, która dla danego drzewa BST zwraca wskaźnik do wierzchołka który jest korzeniem jej maksymalnego zbalansowanego poddrzewa. (Drzewo BST jest *zbalansowane* gdy różnica rozmiarów jego poddrzew jest co najwyżej 1.

Zadanie 29

Napisać procedurę w języku Pascal, która dla danego drzewa BST o kluczach całkowitych, znajduje wskaźnik do korzenia maksymalnego poddrzewa, o kluczach parzystych.

Zadanie 30

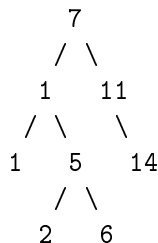
Napisać funkcję w języku Pascal, która dla danego drzewa binarnego zwraca liczbę wierzchołków które są korzeniami pełnych poddrzew binarnych. Np. dla drzewa *d*



funkcja powinna zwrócić wartość 11 ponieważ wierzchołki o etykietach: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12 są korzeniami pełnych poddrzew binarnych.

Zadanie 31

Napisać funkcję w języku Pascal, która dla danego drzewa BST o kluczach całkowitych, tworzy listę kluczy znajdujących się na jednej z najdłuższej gałęzi tego drzewa i zwraca głowę do tej listy. Np. dla drzewa:



procedura powinna zwrócić głowę do listy (7, 1, 5, 6) lub (7, 1, 5, 2)

Grafy

Zadanie 32

Dany jest plik, w liniach którego są różne pary liczb całkowitych z przedziału 1 do n . Napisać procedurę, która tworzy graf o n wierzchołkach reprezentowany jako listy incydencji taki, że pary z pliku to krawędzie tego grafu. W wersji niezorientowanej listy powinny być dwukierunkowe z wartownikiem i obie reprezentacje tej samej krawędzi powinny mieć wskaźniki do siebie na wzajem.

Zadanie 33

Napisać procedurę w języku Pascal, która dla danego grafu zorientowanego $G = (V, E)$ reprezentowanego jako listy incydencji, oblicza graf transponowany $G^T = (V, E^T)$, gdzie $E^T = \{(v, u) : (u, v) \in E\}$.

Zadanie 34

Napisać procedurę w języku Pascal, która dla danego grafu zorientowanego $G = (V, E)$ reprezentowanego jako listy incydencji, który jest drzewem oblicza jego wysokość G .

Zadanie 35

Napisać procedurę, która sprawdza czy dany graf niezorientowany jest spójny.

Zadanie 36

Napisać procedurę, która sprawdza czy dany graf niezorientowany jest acykliczny.

Zadanie 37

W tablicy A znajduje się permutacja liczb od 1 do n . Napisać procedurę w języku Pascal, która sprawdza czy permutacja z tablicy A reprezentuje cykl Hamiltona w grafie G o n wierzchołkach reprezentowanym przez listy incydencji.

Cykl w grafie G *jest cyklem Hamiltona* jeśli przechodzi przez każdy wierzchołek dokładnie jeden raz.

Zadanie 38

Napisać procedurę w języku Pascal, która sprawdza czy dany graf skierowany reprezentowany przez listy incydencji jest acykliczny.

Zadanie 39

Napisać procedurę w języku Pascal, która sprawdza czy na liście o głowie h znajdują się wierzchołki cyklu Eulera grafu skierowanego G o n wierzchołkach.