Zadania przykładowe do pierwszego kolokwium z AA

Zadanie 1

Rozwiąż metodą czynnika sumacyjnego układ równań (wyznacz $T(n)$ jako funkcję n):

$$\begin{cases} T(0) &= 3 & \text{dla} \quad n = 0 \\ 3T(n) &= nT(n-1) + 2n! & \text{dla} \quad n > 0 \end{cases}$$

Zadanie 2

Rozwiąż metodą czynnika sumacyjnego układ równań (wyznacz $T(n)$ jako funkcję n):

$$\begin{cases} T(n) &= 2T(n-1) - 9 \\ T(0) &= 1 \end{cases}$$

Zadanie 3

Rozwiąż metodą czynnika sumacyjnego układ równań (wyznacz $T(n)$ jako funkcję n):

$$\begin{cases} 3T(n-1) &= -4(n) - 5 \\ T(0) &= 1 \end{cases}$$

Zadanie 4

Rozwiąż metodą czynnika sumacyjnego układ równań (wyznacz $T(n)$ jako funkcję n):

$$\begin{cases} T(0) &= 1 \\ T(n) &= n(n-1)T(n-1) + 2\prod_{k=1}^{n} k^2 \end{cases}$$

Zadanie 5

Rozwiąż metodą czynnika sumacyjnego układ równań (wyznacz $T(n)$ jako funkcję n):

$$\begin{cases} T(n) &= 3(T(n-1) - 1) \\ T(0) &= 1 \end{cases}$$

Zadanie 6

Rozwiąż metodą czynnika sumacyjnego układ równań (wyznacz $T(n)$ jako funkcję n):

$$\begin{cases} 3T(n) &= 2T(n-1) + n \\ T(0) &= 2 \end{cases}$$

Zadanie 7

Rozwiąż metodą czynnika sumacyjnego następujące równanie rekurencyjne:

$$\begin{cases} T(0) & = & 0 & \text{dla } n = 0 \\ 2nT(n) & = & n^2 T(n-1) + \sum\limits_{i=1}^{n} n! & \text{dla } n > 0 \end{cases}$$

Zadanie 8

Rozwiąż metodą czynnika sumacyjnego następujące równanie rekurencyjne:

$$T_r(n) \quad = \quad \sum_{1 \leqslant i \leqslant n} \left(\frac{1}{n} \left(n - 1 + \frac{i-1}{n} T_r(i-1) + \frac{n-i}{n} T_r(n-i) \right) \right).$$

Zadanie 9

Stosując metodę zaburzania, oblicz wartość:

$$\sum_{0 \leqslant k \leqslant n} k^2.$$

Zadanie 10

Wyznacz średnią liczbę porównań x z elementami tablicy a dla poniższego algorytmu wyszukiwania binarnego. Załóż, że wyszukiwany element występuje w tablicy a dokładnie raz. Przyjmij, że procedura jest wywoływana z parametrami $l = 1, r = 31$.

```
1  procedure BinarySearch(a, l, r, x : integer);
2  var  s : integer
3  begin
4      s := (l + r) div 2;
5      if a[s] = x then
6          return s;
7      else
8          if a[s] > x then
9              return BinarySearch(a, l, s − 1, x);
10         else
11             return BinarySearch(a, s + 1, r, x);
12         end if;
13     end if;
14 end.
```

Zadanie 11

Wyznacz zależność liczby operacji dominujących od parametrów dla następującej procedury:

```
1  procedure powieksz(sx, sy, skala:integer);
2  var  i, j, k, l:integer;
3  begin
       {sx > 0, sy > 0, skala > 1}
4      i := sx − 1;
5      while i >= 0 do
6          j := sy − 1;
7          while j >= 0 do
8              if GetPixel(i, j) = White then
9                  for k := 0 to skala − 1 do
10                     for l := 0 to skala − 1 do
11                         PutPixel(skala ∗ i + k, skala ∗ j + l, White);
12                 end for;
13             end for;
14         end if;
15         PutPixel(i, j, Black);
16         j := j − 1;
17     end while;
18     i := i − 1;
19 end while;
20 end.
```

Załóż, że operacjami dominującymi są odwołania do biblioteczki graficznej, tzn. wywołania funkcji `GetPixel` i procedury `PutPixel`. Załóż, że funkcja `GetPixel` zwraca tylko `White` lub `Black` z jednakowym prawdopodobieństwem.

Zadanie 12

Dana jest następująca procedura:

```
1  procedure x(n);
2  begin
3      a := random(1, n); b := random(1, n + 2); c := random(3, n + 4); d := random(4, n + 3);
4      e := 0;
5      if c ⩽ a then
6          e := e + 1;
7      end if;
8      if d ⩽ b then
9          e := e + 1;
10     end if;
11 end.
```

Funkcja `random`(d, g) zwraca losową liczbę całkowitą z przedziału $[d, g]$. Wyznacz wartość oczekiwaną e przy wyjściu z procedury.

Zadanie 13

Dla algorytmu `sortowanie_szybkie1` zaproponuj ciąg dziesięciu elementów o takiej własności, że w każdym kroku, jeśli podtablica do sortowania ma długość n większą niż 2, to jest ona dzielona na tablice o długości $n - 3$ oraz 2 (dwa największe elementy). Wykonaj dla takiego ciągu algorytm sortowania.

```
1  procedure sortowanie_szybkie1(d, g);
2  begin
3      if d < g then
4          t := A[d];      {t jest kluczem osiowym}
5          s := d;
6          for i := d + 1 to g do      {przemieszczanie elementów wokół klucza osiowego}
7              if A[i] < t then
8                  s := s + 1;
9                  zamiana(A[s], A[i]);
10             end if;
11         end for;
12         zamiana(A[d], A[s]);
13         sortowanie_szybkie1(d, s − 1);      {wywołania rekursywne dla obu części tablicy}
14         sortowanie_szybkie1(s + 1, g);
15     end if;
16 end.
```

Zadanie 14

Dla algorytmu sortowania bąbelkowego:

```
1  procedure sortowanie_bąbelkowe;
2  begin
3      for i := 1 to n do
4          for j := 2 to n do
5              if A[j − 1] > A[j] then
6                  zamiana(A[j − 1], A[j]);
7              end if;
8          end for;
9      end for;
10 end.
```

1. wyznacz dokładną złożoność algorytmu w przypadku pesymistycznym i średnim (za operację dominującą przyjmij porównanie elementów tablicy),
2. zaproponuj modyfikację algorytmu pozwalającą na zmniejszenie o około połowę złożoności oraz oblicz dla zmodyfikowanego algorytmu jego złożoność,
3. zaproponuj, jak można jeszcze poprawić złożoność tego algorytmu i oszacuj jak duży będzie zysk.

Zadanie 15

Wyznacz złożoność algorytmu `sortowanie_minmax` dla następujących algorytmów wyznaczania elementu minimalnego i maksymalnego: (a) `minmax1`, (b) `minmax2`, (c) `minmax3`.
Uwaga: treści wszystkich algorytmów wziąć z ćwiczeń.

Zadanie 16

Dla podanego ciągu kluczy $20, 8, 15, 9, 4, 22, 17, 26, 10, 21$ zbuduj drzewo poszukiwań binarnych ilustrując kolejne kroki. Następnie usuń wszystkie elementy w takiej kolejności: $4, 20, 8, 10, 15, 9, 22, 17, 26, 21$ również ilustrując wykonywane operacje.

Zadanie 17

Pewien procesor z ograniczonym zestawem instrukcji potrafi tylko dodawać i odejmować jedynkę od liczb całkowitych. Dlatego dodawanie i mnożenie liczb jest dosyć kłopotliwe i musi być realizowane takimi funkcjami:

```
1  function dodaj(a, b : integer) : integer;
2  begin
3      if b = 0 then
4          dodaj := a;
5      else
6          dodaj := 1+dodaj(a, b − 1);      { « zaznaczony wiersz, jedno dodawanie}
7      end if;
8  end.
```

```
1  function mnoz(a, b : integer) : integer;
2  begin
3      if b = 1 then
4          mnoz := a;
5      else
6          mnoz :=dodaj(a, mnoz(a, b − 1));
7      end if;
8  end.
```

Wyznacz zależność liczby dodawań w zaznaczonym wierszu przy mnożeniu `mnoz(`x, y`)` od x i y. (Liczymy oczywiście tylko `+` w zaznaczonym wierszu.)
Wskazówka: przeanalizuj działanie algorytmu dla niewielkich liczb, np. `mnoz(2,3)`.

Zadanie 18

Udowodnij metodą indukcji matematycznej poprawność poniższego wzoru:

$$1^2 + 3^2 + 5^2 + \cdots + (2n-1)^2 = \frac{n(4n^2-1)}{3}$$

Zadanie 19

Wyznacz średnią liczbę porównań x z elementami tablicy a dla poniższego algorytmu wyszukiwania binarnego. Załóż, że tablica A jest posortowana rosnąco, a szukany element x występuje w niej dokładnie raz. Przyjmij, że procedura jest wywoływana z parametrami $d = 1$, $g = n$. Należy założyć, że $n = 2^k - 1$.

```
1  procedure BinarySearch(a, l, r, x : integer);
2  var  s : integer
3  begin
4      s := (l + r) div 2;
5      if a[s] = x then
6          return s;
7      else
8          if a[s] > x then
9              return BinarySearch(a, l, s − 1, x);
10         else
11             return BinarySearch(a, s + 1, r, x);
12         end if;
13     end if;
14 end.
```

Zadanie 20

Wyznacz zależność liczby operacji dominujących od parametru n dla następującej procedury:

```
1  procedure rysuj(n: integer);
2  begin
3      if n = 1 then
4          Forwd(1);
5      else
6          rysuj(n − 1);
7          TurnLeft(60);
8          rysuj(n − 1);
9          TurnRight(120);
10         rysuj(n − 1);
11         TurnLeft(60);
12         rysuj(n − 1);
13     end if;
14 end.
```

Załóż, że operacjami dominującymi są odwołania do biblioteki graficznej, tzn. wywołania procedur `Forwd`, `TurnLeft` i `TurnRight`. Należy przyjąć, że procedura `rysuj` jest wywoływna z parametrem $n > 0$.

Zadanie 21

Wyznacz złożoność poniższego algorytmu (wyznacz ją w zależności od argumentu n). Pierwsze wywołanie wygląda następująco: `proc(`n`)`. Za operację dominującą przyjmij wywołanie procedury `rysuj`.

```
1  procedure proc(n : integer);
2  begin
3      if n = 0 then
4          rysuj(n);
5          return;
6      end if;
7      if n > 3 then
8          proc(n − 1);
9      end if;
10     proc(n − 1);
11 end.
```

Zadanie 22

Przedstaw proces sortowania algorytmem `sortowanie_szybkie1` dla następującej tablicy:
$[15, 8, 4, 2, 6, 9, 13, 7, 10, 1]$.

```
1  procedure sortowanie_szybkie1(d, g);
2  begin
3      if d < g then
4          t := A[d];      {t jest kluczem osiowym}
5          s := d;
6          for i := d + 1 to g do      {przemieszczanie elementów wokół klucza osiowego}
7              if A[i] < t then
8                  s := s + 1;
9                  zamiana(A[s], A[i]);
10             end if;
11         end for;
12         zamiana(A[d], A[s]);
13         sortowanie_szybkie1(d, s − 1);      {wywołania rekursywne dla obu części tablicy}
14         sortowanie_szybkie1(s + 1, g);
15     end if;
16 end.
```

Zadanie 23

Daną wejściową poniższego algorytmu jest całkowita liczba n losowana z przedziału $[10000, 19999]$ (wszystkie liczby są jednakowo prawdopodobne). Wyznacz oczekiwaną liczbę wywołań procedury `PutPixel`.

```
1  procedure procedure rysuj(n);
2  var  q, x : integer
3  begin
4      q := n mod 10;
5      x := 0;
6      while n <> 0 do
7          if n mod 10 = q then
8              if n mod 3 < 2 then
9                  PutPixel(x, q);
10                 PutPixel(x, q + 1);
11             end if;
12         else
13             PutPixel(x, q);
14         end if;
15         x := x + 1;
16         n := n div 10;
17     end while;
18 end.
```

Zadanie 24

Wyznacz złożoność poniższego algorytmu (wyznacz ją w zależności od argumentu n). Pierwsze wywołanie wygląda następująco: `proc(`n`)`. Za operację dominującą przyjmij wywołanie procedury `rysuj`.

```
1  procedure proc(n : integer);
2  begin
3      if n = 0 then
4          rysuj();
5          rysuj();
6          return;
7      end if;
8      if n > 2 then
9          proc(n − 1);
10     end if;
11     proc(n − 1);
12     proc(n − 1);
13 end.
```

Zadanie 25

W tablicy A[0..n-1] zapisany jest pewien ciąg zer i jedynek (tzn. $A[i] = 0$ lub $A[i] = 1$). Dany jest następujący algorytm:

```
1  procedure incr;
2  begin
3      i := 0;
4      while i < nand A[i] = 0 do
5          A[i] := 0;
6          i := i + 1;
7      end while;
8      if i < n then
9          A[i] := 1;
10     end if;
11 end.
```

Jaki jest średni czas działania algorytmu w zależności od n, jeśli jako dominującą operację przyjmiemy zmianę elementu tablicy A, a każdy ciąg zer i jedynek jest jednakowo prawdopodobny?

Zadanie 26

Wyznacz zależność liczby operacji dominujących od parametrów dla następującej procedury:

```
1  procedure powieksz(sx, sy, skala:integer);
2  var  i, j, k, l:integer;
3  begin
       {sx > 0, sy > 0, skala > 1}
4      i := sx − 1;
5      while i >= 0 do
6          j := sy − 1;
7          while j >= 0 do
8              if GetPixel(i, j) = White then
9                  for k := 0 to skala − 1 do
10                     for l := 0 to skala − 1 do
11                         PutPixel(skala ∗ i + k, skala ∗ j + l, White);
12                 end for;
13             end for;
14         end if;
15         PutPixel(i, j, Black);
16         j := j − 1;
17     end while;
18     i := i − 1;
19 end while;
20 end.
```

Załóż, że operacjami dominującymi są odwołania do biblioteczki graficznej, tzn. wywołania funkcji `GetPixel` i procedury `PutPixel`. Załóż, że funkcja `GetPixel` zwraca tylko `White` lub `Black` z jednakowym prawdopodobieństwem.

Zadanie 27

W tablicach $A1, A2[0..n-1]$ zapisane są dwa ciągi znaków ($'a' \leqslant Ax[i] \leqslant 'z'$). Dany jest następujący algorytm:

```
1  function cmp : boolean;
2  begin
3      i := 0;
4      while (A1[i] = A2[i])and (i < n) do
5          i := i + 1;
6      end while;
7      return (i = n);
8  end.
```

Jaki jest średni czas działania algorytmu w zależności od n, jeśli jako dominującą operację przyjmiemy porównywania elementów tablic $A1$ i $A2$? Każdy ciąg znaków jest jednakowo prawdopodobny.

Zadanie 28

Wyznacz dokładną, średnią liczbę wykonywanych porównań elementu x z elementami tablicy A w trakcie działania funkcji procedury wyszukiwanie_binarne. Załóż, że tablica A jest posortowana rosnąco, a szukany element x występuje w niej dokładnie raz. Przyjmij, że procedura jest wywoływana z parametrami $d = 1$, $g = n$. Dla uproszczenia przyjmij, że $n = 2^k - 1$.

```
1  function wyszukiwanie_binarne(A, d, g, x);
2  begin
3      k := (d + g) div 2;
4      if A[k] = x then
5          return k;
6      else
7          if A[k] > x then
8              return wyszukiwanie_binarne(A, d, k − 1, x);
9          else
10             return wyszukiwanie_binarne(A, k + 1, g, x);
11         end if;
12     end if;
13 end.
```

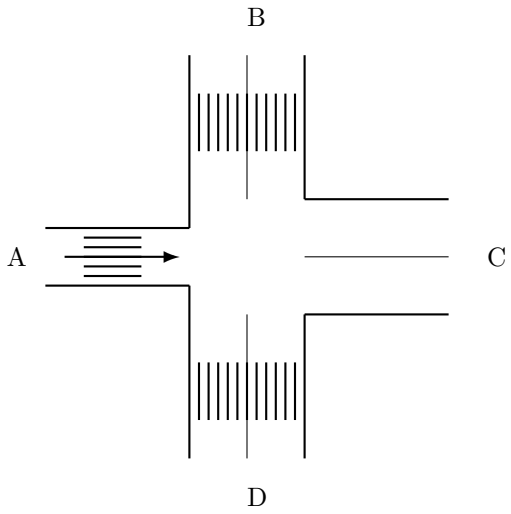
Zadanie 29

Dla podanego poniżej algorytmu sortowania Shella, zilustruj proces sortowania tablicy liczb $\{5, 13, 8, 17, 65, 12, 9, 41, 25, 1, 14\}$.

```
1  procedure sortowanie_shella;
2  begin
3      h := 1;
4      while h < n/9 do      {wyszukiwanie maksymalnego h}
5          h := 3 * h + 1;
6      end while;
7      while h > 0 do      {wykonuj h-sortowania tak długo, aż h zmaleje do 0}
8          for i := h + 1 to n do
9              x := A[i]; j := i;
10             while j ⩾ h + 1 cand x < A[j − h] do
11                 A[j] := A[j − h]; j := j − h;
12             end while;
13             A[j] := x;
14         end for;
15         h := h div 3;      {zmniejszenie wartości h}
16     end while;
17 end.
```

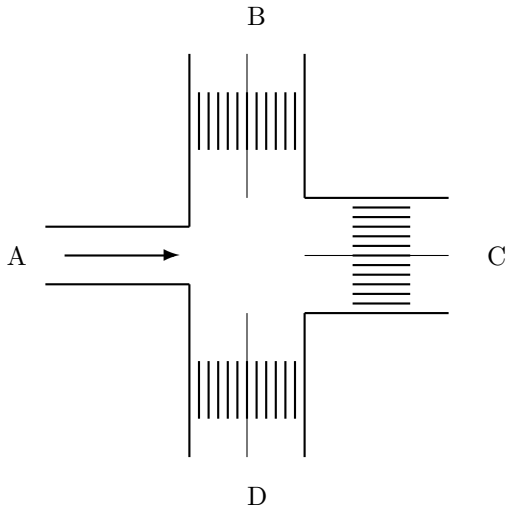
Zadanie 30

Zaprojektuj sekwencję zmiany świateł na przedstawionym poniżej skrzyżowaniu wykorzystując kolorowanie grafu algorytmem zachłannym. Uwzględnij zaznaczone przejścia dla pieszych.



Zadanie 31

Zaprojektuj sekwencję zmiany świateł na przedstawionym poniżej skrzyżowaniu wykorzystując kolorowanie grafu algorytmem zachłannym. Uwzględnij zaznaczone przejścia dla pieszych.



Zadanie 32

Dla podanego niżej algorytmu sortowania, wyznacz złożoność w przypadku średnim i pesymistycznym. Za operację dominującą przyjmij porównanie elementów tablicy A. Załóż, że n jest liczbą parzystą.

```
1  procedure sortowanie;
2  begin
3      d := 1; g := n;
4      while d < g do
5          m := d; M := d;
6          for i := d + 1 to g do
7              if A[i] < A[m] then
8                  m := i;
9              end if;
10             if A[i] > A[M] then
11                 M := i;
12             end if;
13         end for;
14         zamiana(A[d], A[m]);
15         zamiana(A[g], A[M]);
16         d := d + 1;
17         g := g − 1;
18     end while;
19 end.
```

Zadanie 33

Zmodyfikujmy grę w kółko i krzyżyk na planszy $3 * 3$ w następujący sposób:

- grę rozgrywamy zawsze do końca,
- po wypełnieniu planszy obliczamy liczbę wystąpień krzyżyka we wszystkich wierszach i kolumnach,
- krzyżyk wygrywa wtedy i tylko wtedy gdy sumy wystąpień krzyżyka w odpowiadających sobie wierszach i kolumnach (pierwszy wiersz – pierwsza kolumna, drugi wiersz – druga kolumna, trzeci wiersz – trzecia kolumna) są identyczne,
- kółko wygrywa w przeciwnym wypadku.

Przykłady:

X	X	O	2
X	O	X	2
O	X	O	1
2	2	1	**X**

X	O	O	1
X	O	O	1
X	X	X	3
3	1	1	**O**

Dla tak zmodyfikowanej gry, oceń poniższe pozycje stosując algorytm minimaxa (stwierdź, który z graczy wygra, jeśli obydwaj będą grali optymalnie). Kolejny ruch przypada na krzyżyk. Narysuj pełne, od bieżącej pozycji w dół, drzewo gry.

O	X	O
X	X	
O		

O		O
X	O	X
	X	

Zadanie 34

Dany jest graf nieskierowany $G = (V, E)$. Z każdą krawędzią $(u, v) \in E$ związana jest pewna waga będąca liczbą rzeczywistą. Zaproponuj algorytm wyszukiwania w tym grafie takiego cyklu, w którym największa spośród wag jego krawędzi jest minimalna. Wynikiem działania algorytmu powinna być ta największa waga. Oszacuj złożoność zaproponowanego algorytmu.

Zadanie 35

Dany jest graf nieskierowany $G = (V, E)$. Z każdą krawędzią $(u, v) \in E$ związana jest pewna waga będąca liczbą rzeczywistą. Zaproponuj algorytm wyszukiwania w tym grafie takiego cyklu, w którym najmniejsza spośród wag jego krawędzi jest maksymalna. Wynikiem działania algorytmu powinna być ta najmniejsza waga. Oszacuj złożoność zaproponowanego algorytmu.

Zadanie 36

Zmodyfikujmy grę w kółko i krzyżyk na planszy $3 * 3$ w następujący sposób:

- grę rozgrywamy zawsze do końca,
- po wypełnieniu planszy obliczamy liczbę wystąpień krzyżyka we wszystkich wierszach i kolumnach,
- krzyżyk wygrywa wtedy i tylko wtedy gdy sumy wystąpień krzyżyka w odpowiadających sobie wierszach i kolumnach (pierwszy wiersz – pierwsza kolumna, drugi wiersz – druga kolumna, trzeci wiersz – trzecia kolumna) są identyczne,
- kółko wygrywa w przeciwnym wypadku.

Przykłady:

X	X	O	2
X	O	X	2
O	X	O	1
2	2	1	**X**

X	O	O	1
X	O	O	1
X	X	X	3
3	1	1	**O**

Dla tak zmodyfikowanej gry, oceń poniższe pozycje stosując algorytm minimaxa (stwierdź, który z graczy wygra, jeśli obydwaj będą grali optymalnie). Kolejny ruch przypada na krzyżyk. Narysuj pełne, od bieżącej pozycji w dół, drzewo gry.

X	X	O
X	O	
O		

	X	O
X		X
O		O

Zadanie 37

Dane są funkcje mieszające:

$$\begin{cases} h_0(x) = x \ mod \ 8 \\ h_i(x) = h_0(x) + 2i^2 - 5i \qquad \text{dla } 1 \leqslant x \leqslant 7 \end{cases}$$

Przedstaw postać tablicy mieszającej (mieszanie zamknięte) po wprowadzeniu do niej liczb $57, 21, 18, 5, 123, 87, 25, 33$. Wyznacz wykonaną liczbę prób wstawiania elementów. Dla porównania przedstaw postać tablicy mieszającej (mieszanie otwarte) przy zastosowaniu funkcji mieszającej h_0. Nowe elementy wstawiamy zawsze na początek list.

Zadanie 38

Dane są funkcje mieszające:

$$\begin{cases} h_0(x) = x \ mod \ 8 \\ h_i(x) = h_0(x) + 2i^2 - 5i \qquad \text{dla } 1 \leqslant x \leqslant 7 \end{cases}$$

Przedstaw postać tablicy mieszającej (mieszanie zamknięte) po wprowadzeniu do niej liczb $18, 21, 33, 5, 123, 87, 57, 25$. Wyznacz wykonaną liczbę prób wstawiania elementów. Dla porównania przedstaw postać tablicy mieszającej (mieszanie otwarte) przy zastosowaniu funkcji mieszającej h_0. Nowe elementy wstawiamy zawsze na początek list.

Zadanie 39

Dane jest następujące zadanie:

```
   ŻONA
   ŻONA
+  ŻONA
   ŻONA
  ------
  HAREM
```

Rozwiązaniem tego problemu jest znalezienie takiego przyporządkowania literom cyfr, żeby suma była poprawna. Każda cyfra może być przyporządkowana tylko jednej literze. Napisz algorytm, rozwiązujący to i podobne zadania. Oszacuj jego złożoność.

Zadanie 40

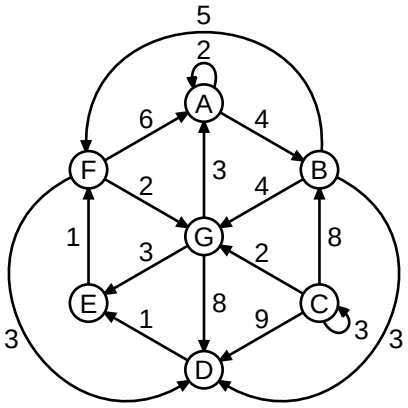
Dane jest następujące zadanie:

```
   5678
   5678
+  5678
   5678
  ------
  22712
```

Rozwiązaniem tego problemu jest znalezienie takiego przyporządkowania liter cyfrom, żeby w sumie występowały tylko prawidłowe słowa. Każda litera może być przyporządkowana tylko jednej cyfrze. Napisz algorytm, rozwiązujący to i podobne zadania. Oszacuj jego złożoność. Załóż, że dana jest funkcja `sprawdź_słowo`, która weryfikuje czy słowo jest poprawne (złożoność tej funkcji to $O(1)$).

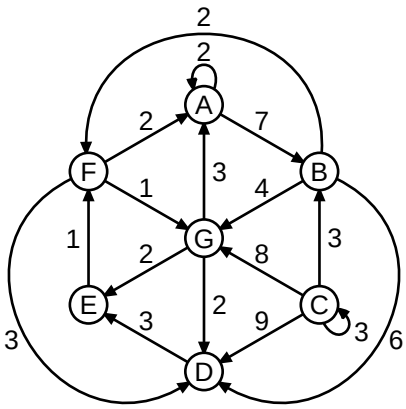
Zadanie 41

Znajdź w poniższym grafie skierowanym długość najkrótszych dróg od wierzchołka C do wszystkich innych wierzchołków. Przedstaw poszczególne kroki znajdowania rozwiązania.



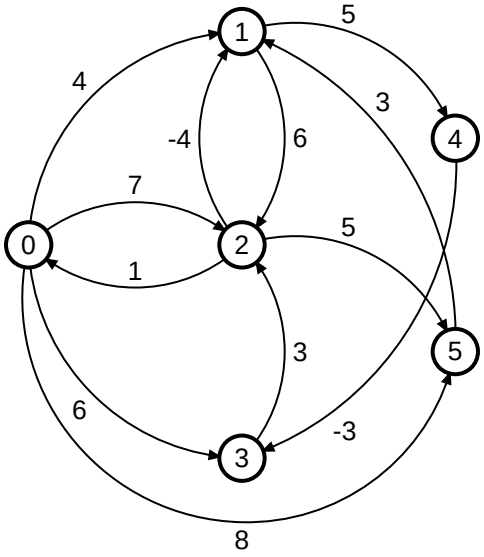
Zadanie 42

Znaleźć w poniższym grafie skierowanym długość najkrótszych dróg od wierzchołka C do wszystkich innych wierzchołków. Przedstawić poszczególne kroki znajdowania rozwiązania.



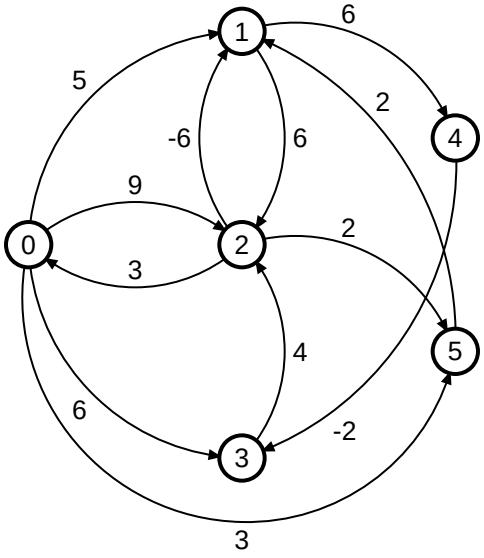
Zadanie 43

W podanym grafie znajdź, za pomocą jednego z poznanych na ćwiczeniach algorytmów (nazwij ten algorytm), koszty najtańszych ścieżek z wierzchołka 0 do wszystkich innych wierzchołków. Przedstaw poszczególne kroki działania zastosowanego algorytmu.



Zadanie 44

W podanym grafie znajdź, za pomocą jednego z poznanych na ćwiczeniach algorytmów (nazwij ten algorytm), koszty najtańszych ścieżek z wierzchołka 0 do wszystkich innych wierzchołków. Przedstaw poszczególne kroki działania zastosowanego algorytmu.



Zadanie 45

Dany jest zbiór $\{A, B, C, D, E, F, G, H\}$. Wygeneruj 29–tą kombinację pięcioelementową oraz 17–tą kombinację trójelementową. Zilustruj procedurę wyboru.

Zadanie 46

Profesor Midas jedzie samochodem z Newark do Reno. Bak pełen benzyny w jego bolidzie wystarcza na przejechanie n km, a na jego mapie są zaznaczone odległości między wszyskimi stacjami benzynowymi na trasie. Profesor ma zamiar tankować benzynę jak najmniejszą liczbę razy w trakcie podróży. Podaj algorytm, który profesor Midas może wykorzystać aby ustalić, na których stacjach powinien tankować. Liczba stacji wynosi s. Oszacuj złożoność tego algorytmu.

Zadanie 47

Dana jest następująca gra. Dostępny jest stos kart, S, zawierających liczby od 1 do 9. Dwóch graczy, A i B, wykonuje naprzemiennie ruchy (zaczyna gracz A). Ruchem gracza może być:

- wzięcie karty znajdującej się na szczycie stosu,
- zrezygnowanie z wzięcia tej karty.

Jeżeli obydwaj gracze bezpośrednio po sobie zrezygnują z wzięcia karty, to gra się kończy. Gra kończy się również wtedy, gdy stos się wyczerpie. Po zakończeniu gry, gracze sumują liczby na kartach, które wzięli otrzymując sumy S_A oraz S_B. Wynikiem gry jest różnica $R = S_A - S_B$, która jest wypłatą przekazywaną graczowi A przez gracza B. Jeśli R jest liczbą ujemną, to oczywiście A wypłaca B kwotę $|R|$. Każdy z graczy dąży do zmaksymalizowania swojej wypłaty, tzn. A stara się, żeby R była jak największa, a B aby R była jak najmniejsza.

Przykład 1: Początkowa zawartość stosu to $S = (2, 4, 1)$. Przykładowy przebieg partii:

- A bierze kartę ze szczytu czyli 2 (zawartość stosu: $S = (4, 1)$),
- B bierze kartę ze szczytu czyli 4 (zawartość stosu: $S = (1)$),
- A bierze kartę ze szczytu czyli 1 (zawartość stosu: $S = ()$),
- gra się kończy ze względu na opróżnienie stosu.

W tym wypadku $S_A = 2 + 1 = 3$, $S_B = 4$, $R = S_A - S_B = 3 - 4 = -1$, a więc gracz A wypłaca graczowi B kwotę 1.

Przykład 2: Początkowa zawartość stosu to $S = (2, 4, 1)$. Przykładowy przebieg partii:

- A bierze kartę ze szczytu czyli 2 (zawartość stosu: $S = (4, 1)$),
- B rezygnuje z wzięcia karty (zawartość stosu: $S = (4, 1)$),
- A rezygnuje z wzięcia karty (zawartość stosu: $S = (4, 1)$),
- gra się kończy ze względu na to, że gracze bezpośrednio po sobie zrezygnowali z wzięcia karty.

W tym wypadku $S_A = 2$, $S_B = 0$, $R = S_A - S_B = 2 - 0 = 2$, a więc gracz B wypłaca graczowi A kwotę 2.

Dla tak zdefiniowanej gry narysuj pełne drzewa gry i zakładając optymalną grę obydwu graczy (zastosuj algorytm minimax) oblicz wartość gry (podaj jakim wynikiem gra się skończy jeśli obydwaj gracze będą grali optymalnie) dla początkowych zawartości stosu:

1. $S = (2, 4, 1)$,
2. $S = (3, 5, 3)$.

Działanie algorytmu minimax powinno zostać zilistrowane na drzewach gry.

Zadanie 48

Przedstaw proces znajdowania 13-tej i 19-tej kombinacji czteroelementowej ze zbioru $\{A, B, C, D, E, F, G\}$.

Zadanie 49

Przedstaw 3 różne drzewa poszukiwań binarnych, z których każde zawiera liczby: $11, 8, 7, 15, 21, 4, 25, 41, 26, 24$.

Zadanie 50

Zbuduj drzewo Huffmana dla następujących symboli: $a(5)$, $b(3)$, $c(2)$, $d(7)$, $e(3)$, $f(6)$, $g(12)$. W nawiasach podano częstość występowania symbolu.

Zadanie 51

Przedstaw proces wyszukiwania 7-go i 10-go co do wielkości elementu w tablicy $[15, 9, 67, 88, 91, 33, 92, 41, 45, 5, 75, 49, 8, 35, 56]$. Zastosuj algorytm o średniej złożoności liniowej, którego ideę omawiano na ćwiczeniach.

Zadanie 52

Przedstaw proces sortowanie przez kopcowanie tablicy liczb: $[15, 3, 83, 153, 4, 8, 9, 7, 21]$.

Zadanie 53

Przedstaw proces budowania drzewa poszukiwań binarnych zawierających liczby: $15, 3, 83, 153, 4, 8, 9, 7, 21, 22$. Następnie przedstaw proces usuwania tych liczb w takiej samej kolejności, w jakiej były one wstawiane.

Zadanie 54

Zilustruj proces wyszukiwania NWP ciągów $X = \langle A, B, B, D, A, B, C\rangle$ i $Y = \langle B, C, A, B, B, D, A\rangle$ za pomocą algorytmu poznanego na ćwiczeniach.

Zadanie 55

Dla podanego ciągu kluczy $16, 23, 15, 5, 17, 12, 25, 21, 20, 22$ zbuduj drzewo poszukiwań binarnych przedstawiając jego postać końcową. Następnie usuń wszystkie elementy w kolejności: $16, 23, 25, 17, 20, 15, 21, 5, 22, 12$. Należy przedstawić postać drzewa po usunięciu każdego z elementów!

Zadanie 56

Dane są funkcje mieszające:

$$\begin{cases} h(x,0) = x \bmod 8 \\ h(x,i) = (h(x,0) + 5i) \bmod 8 \qquad \text{dla } 1 \leqslant i \leqslant 7 \end{cases}$$

Przedstaw postać tablicy mieszającej (mieszanie zamknięte) po wprowadzeniu do niej liczb $14, 28, 9, 0, 35, 20, 6$. Dla porównania przedstaw postać tablicy mieszającej (metoda łańcuchowa) przy zastosowaniu funkcji mieszającej $h(x, 0)$.

Zadanie 57

Dla podanej macierzy odległości między miastami wykonaj algorytm wyznaczający marszrutę komiwojażera. Użyj algorytmu zachłannego, który był przedstawiony na ćwiczeniach.

	a	b	c	d	e	f
a	-	4,24	10,82	8,54	6,71	3,00
b		-	12,37	11,00	3,00	3,00
c			-	3,16	12,00	9,49
d				-	11,40	8,00
e					-	4,24
f						-