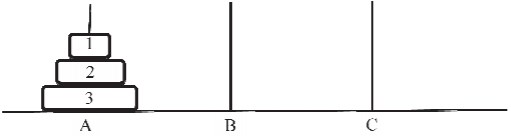
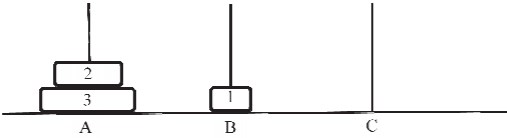
**Analiza algorytmów – Problem Wieży z Hanoi**

**Imię, nazwisko, klasa:**

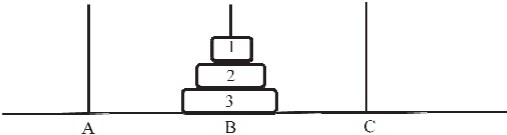
W problemie wież z Hanoi mamy trzy pręty oznaczone A, B i C oraz n okrągłych krążków o średnicach odpowiednio 1, 2, …, n. Na początku wszystkie krążki nałożone są na pręt A, w kolejności od największego do najmniejszego (największy na dole, najmniejszy na górze). Układ ten (dla n = 3) został przedstawiony na poniższym rysunku.



Zgodnie z regułami problemu krążki można przekładać między prętami. W jednym ruchu możliwe jest przełożenie krążka znajdującego się na szczycie jednego z prętów na szczyt innego pręta, pod warunkiem że nie kładziemy przekładanego krążka na krążek mniejszy od niego. Na przykład na poniższym rysunku krążek 2 możemy przełożyć z pręta A na pręt C, natomiast niemożliwe jest przełożenie go na pręt B.



Zadanie polega na przełożeniu wszystkich krążków z pręta A na pręt B, przy czym można korzystać z pomocniczego pręta C. Na poniższym rysunku przedstawiono efekt końcowy.



Problem wież z Hanoi można rozwiązać za pomocą algorytmu rekurencyjnego. W algorytmie pręty: startowy, docelowy i pomocniczy, podane są jako parametry wejściowe, odpowiednio x, y i z. Algorytm polega na tym, że najpierw przenosimy n – 1 krążków na pręt pomocniczy z, potem największy krążek zostaje przeniesiony na pręt docelowy y, a na koniec n – 1 krążków zostaje przeniesionych z pręta pomocniczego z na pręt docelowy y, przy czym pręt startowy x traktowany jest jako pomocniczy.

**Specyfikacja algorytmu**:

Dane:

*n* — liczba całkowita dodatnia,

*x* — nazwa pręta startowego,

*y* — nazwa pręta docelowego,

*z* — nazwa pręta pomocniczego.

Wynik:

*ciąg ruchów opisujący rozwiązanie problemu wież z Hanoi z n krążkami, w którym na początku wszystkie krążki znajdują się na pręcie x, a na końcu mają znaleźć się na pręcie y, zaś pomocniczym prętem jest z.*

**Uwaga**: Pojedynczy ruch zapisujemy za pomocą znaku =>. Na przykład C => B oznacza przeniesienie krążka z pręta C na pręt B.

**funkcja** *wieże*(*n*, *x*, *y*, *z*)

**jeżeli** *n* = 1

**wypisz** *x* => *y*

**w przeciwnym razie**

*wieże*(*n* – 1, *x*, *z*, *y*)

**wypisz** *x* => *y*

*wieże*(*n* – 1, *z*, *y*, *x*)

**Przykład**

Wywołanie *wieże*(2, A, B, C) spowoduje dwa wywołania rekurencyjne: *wieże*(1, A, C, B) oraz *wieże*(1, C, B, A). Ciąg ruchów utworzony przez *wieże*(2, A, B, C) ma postać:

A => C, A => B, C => B,

gdzie podkreślone ruchy są utworzone przez rekurencyjne wywołania *wieże*(1, A, C, B) oraz *wieże*(1, C, B, A).

**Zadanie 1**

Podaj wszystkie wywołania rekurencyjne funkcji wieże (wraz z ich parametrami), do których dojdzie w wyniku wywołania wieże (3, A, B, C). Odpowiedź podaj w poniższej tabeli, uzupełniając parametry wszystkich wywołań rekurencyjnych.

|  |  |  |  |
| --- | --- | --- | --- |
| *n* | *x* | *y* | *z* |
| 3 | A | B | C |
| 2 | A | C | B |
| 1 | A | B | C |
| 1 |  |  |  |
| 2 |  |  |  |
| 1 |  |  |  |
| 1 |  |  |  |

**Zadanie 2**

Prześledź działanie *wieże*(3, A, B, C) i uzupełnij poniżej wygenerowany ciąg ruchów:

A => B; A => C;.......

Odpowiedź:

**Zadanie 3**

Niech H(*n*) oznacza liczbę ruchów wykonanych przez podany algorytm dla *n* krążków. Zauważ, że rozwiązanie problemu dla n > 1 krążków wymaga jednego ruchu oraz dwukrotnego rozwiązania problemu dla *n* – 1 krążków. W oparciu o tę obserwację uzupełnij poniższą tabelę.

|  |  |
| --- | --- |
| *n* | H(*n*) |
| 1 | 1 |
| 2 | 3 |
| 3 |  |
| 4 |  |
| 5 |  |
| 7 |  |
| 10 |  |

Podaj ogólny wzór określający liczbę ruchów dla n krążków:

Odpowiedź: H(n)=

**Zadanie 4**

Poniżej znajduje się nierekurencyjne rozwiązanie problemu wież z Hanoi:

**Specyfikacja**:

Dane:

*n* — liczba całkowita dodatnia,

Wynik:

*ciąg ruchów opisujący rozwiązanie problemu wież z Hanoi z n krążkami, w którym na początku wszystkie krążki znajdują się na pręcie A, a na końcu powinny się znaleźć na pręcie B.*

**Algorytm**

(1) **dopóki** (pręt A jest niepusty lub pręt C jest niepusty) **wykonuj**

(2) **jeżeli** *n* jest parzyste:

(3) **przenieś** krążek nr 1 o jedną pozycję w lewo

(4) **w przeciwnym razie**

(5) **przenieś** krążek nr 1 o jedną pozycję w prawo

(6) **przenieś** krążek między prętami, na których nie ma krążka nr 1

W powyższym algorytmie przeniesienie krążka nr 1 o jedną pozycję w prawo oznacza wykonanie jednego z ruchów A => B, B => C lub C => A, tak aby krążek nr 1 został przeniesiony na inny pręt. Analogicznie przeniesienie krążka w lewo oznacza wybranie jednego z ruchów A => C, B => A lub C => B, tak aby krążek nr 1 został przeniesiony na inny pręt.

Ruch w kroku (6) powyższego algorytmu jest określony jednoznacznie, gdyż dopuszczalne jest tylko położenie mniejszego krążka na większym, a nie odwrotnie.

**Przykład**

Dla n = 3 powyższy algorytm wykona następujący ciąg ruchów:

A => B; A => C; B => C; A => B; C => A; C => B; A => B,

gdzie ruchy podkreślone przenoszą krążek nr 1 o jedną pozycję w prawo.

Wypisz ciąg ruchów, który poda powyższy algorytm dla n = 4. Uzupełnij poniższą tabelę:

Podkreśl przesunięcia które przenoszą krążek o 1 pozycje w prawo lub w lewo

W kolumnie „Stan wież po przesunięciu” zastosuj format a – b - c, gdzie a,b,c oznaczają krążki na prętach oraz stanem wejściowym jest 1234 – x – x, gdzie x to brak krążków na danym pręcie.

|  |  |  |
| --- | --- | --- |
| Nr operacji | Przesunięcie krążka | Stan wież po przesunięciu |
| 1 | A => C | 234 - x – 1 |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |