

Algorytmy – rodzaje algorytmów

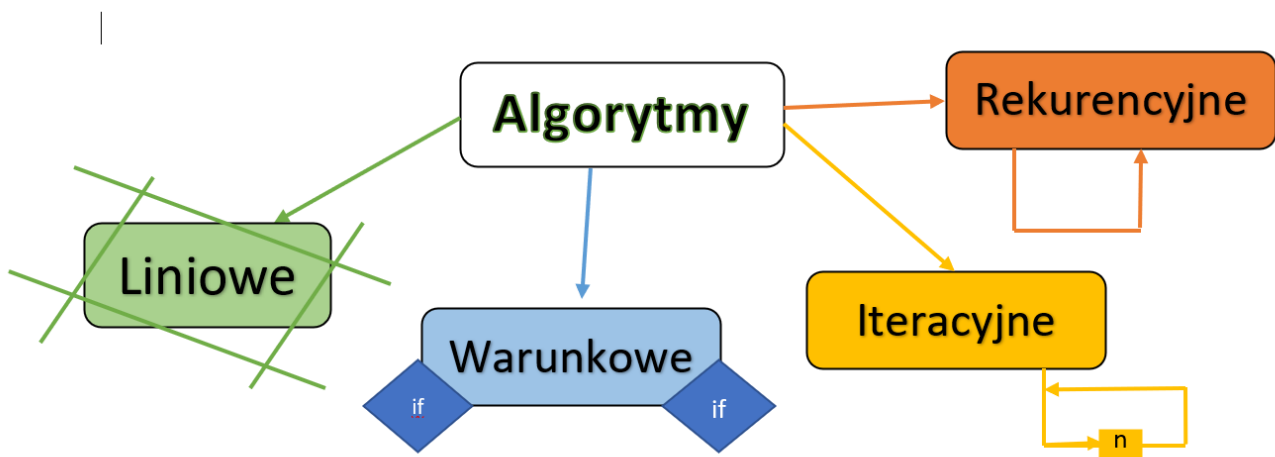
Z tej lekcji dowiesz się:

1. Jakie są rodzaje algorytmów
2. Czym się one charakteryzują
3. Utrwalisz poprzednią lekcję dot. zapisu algorytmów

Algorytmy dzielimy na:

- algorytmy liniowe, czyli algorytmy sekwencyjne
- algorytmy warunkowe, czyli algorytmy z rozgałęzieniami
- algorytmy iteracyjne, czyli algorytmy z pętlą, inaczej algorytmy cykliczne
- algorytmy rekurencyjne

Infografika poniżej dla łatwiejszego zapamiętania i zobrazowania sobie podziału.



Przy wykonywaniu zadań pomocne Wam będą te oto materiały zamieszczone tutaj:

<http://kaluska.pl/schematy-blokowe-algorytmow/>

Algorytm liniowy

Algorytm liniowy to algorytm mający prostą postać. Składa się z ciągu instrukcji, które są wykonywane jedna po drugiej w kolejności, jaka wynika z ich następstwa w zapisie algorytmu. Taki algorytm nosi również nazwę **algorytmu sekwencyjnego**.

Przykład 1. Przykład algorytmu liniowego

Opracuj algorytm telefonicznego zaproszenia koleżanki na ciasto. Numer telefonu : 666-22-88.

W tym przypadku zrobimy założenie, że nic nie utrudni połączenia (a więc już za pierwszym razem zostaniemy połączeni z wybraną osobą), a wykonawca czynności zna cyfry i wie jak wybiera się je z klawiatury telefonu.

Oto jak wygląda algorytm (wersja 1):

1. Podnieś słuchawkę.
2. Wybierz cyfrę 6.
3. Wybierz cyfrę 1.
4. Wybierz cyfrę 6.
5. Wybierz cyfrę 2.
6. Wybierz cyfrę 2.
7. Wybierz cyfrę 2.
8. Wybierz cyfrę 2.
9. Przekaż informacje.
10. Odłóż słuchawkę.

Algorytmy liniowe mają opisy składające się z kroków, które **nie zależą od żadnych warunków** i są wykonywane w zapisanej kolejności.

Zadanie 1. Napisz ten sam algorytm za pomocą schematu blokowego

Algorytmy warunkowe (algorytm rozgałęziony)

Kolejny rodzaj algorytmów to algorytmy warunkowe, zwane inaczej algorytmami rozgałęzionymi.

W rzeczywistości, większość algorytmów nie jest liniowa, ma bardziej rozbudowaną strukturę. Często występują w nich instrukcje, których wykonanie uzależnione jest od spełnienia pewnego **warunku** lub też spełnienie pewnego warunku powoduje wykonanie jednej instrukcji, a niespełnienie go innej. Taką instrukcję nazywamy **instrukcją warunkową**.

Przykład 2. Przykład algorytmu warunkowego

Popraw opracowany wcześniej algorytm tak, aby uwzględnił sytuację, gdy po wybraniu numeru jest on zajęty lub połączenie okazało się błędne.

Kiedy słysząc sygnał zajętości numeru, a więc nie udało się uzyskać połączenia trzeba Odłożyć słuchawkę. Tak samo postępujemy, gdy nawiązane zostało połączenie z innym Numerem.

Aby zrealizować taką sytuację zastosujemy **instrukcję warunkową**. Zrobimy to po to, aby opisać czynności, powinny być wykonane wtedy kiedy zostało nawiązane poprawne połączenie, jak również nie zostało nawiązane. Zauważ, że wtedy wykonawca znajdzie się w punkcie wyjścia, czyli jakby w ogóle nie podjął próby telefonowania.

Algorytm może mieć postać taką (wersja 2):

1. Podnieś słuchawkę.
2. Wybierz cyfrę 6.
3. Wybierz cyfrę 1.
4. Wybierz cyfrę 6.
5. Wybierz cyfrę 2.
6. Wybierz cyfrę 2.
7. Wybierz cyfrę 2.
8. Wybierz cyfrę 2.
9. Czy połączyłeś się z koleżanką ? Jeśli TAK, to przejdź do kroku 10. Jeśli NIE, to przejdź do kroku 11.
10. Zaproś koleżankę.

11. Odłóż słuchawkę.

Zadanie 2. Napisz ten sam algorytm za pomocą schematu blokowego

Algorytmy iteracyjne (algorytm cykliczny)

Kolejny rodzaj algorytmów to algorytm iteracyjny, inaczej algorytm cykliczny lub **algorytm z pętlą**.

Instrukcja iteracyjna – ze znaną ilością powtórzeń

Przyjrzyj się uważnie algorytmowi przedstawionemu wcześniej, powyżej. Zauważyłeś, że istnieją tu powtarzające się instrukcje, aż czterokrotnie występuje „Wybierz cyfrę 2”.

Takie wielokrotne powtarzanie niektórych instrukcji jest cechą charakterystyczną wielu algorytmów, jednak nie zawsze (tak jak w tym algorytmie) możemy określić dokładnie liczbę powtórzeń. Może ona zależeć od spełnienia pewnych warunków. Wielokrotne powtarzanie instrukcji umożliwia instrukcje iteracyjne (pętle). Działa ona według schematu:

Wykonuj instrukcję A dokładnie n razy.

Przykład 4. Przykład algorytmu iteracyjnego.

Popraw opracowany wcześniej algorytm tak, aby sekwencję jednakowych czynności zastąpić pętlą.

Oto algorytm (wersja 3):

1. Podnieś słuchawkę.
2. Wybierz cyfrę 6.
3. Wybierz cyfrę 1.
4. Wybierz cyfrę 6.
5. Wykonaj czynność cztery razy: Wybierz cyfrę 2.
6. Czy połączyłeś się z koleżanką? Jeśli tak, to przejdź do kroku 7. Jeśli nie, to przejdź do kroku 8.
7. Zaproś koleżankę.
8. Odłóż słuchawkę.

Działanie algorytmu może zakończyć się na dwa różne sposoby, albo uzyskamy połączenie z koleżanką i zaprosimy ją na ciasto, albo nie połączymy się i wykonanie algorytmu ograniczy się tylko do wykręcenia numeru.

Jak widzisz zastosowanie **pętli** zmienia sposób zapisu algorytmu, nie zmienia się jednak jego działanie.

Instrukcja iteracyjna – ze spełnieniem warunku

Przykład 5. Przykład algorytmu iteracyjnego warunkowego.

Uwzględnij w opracowanym wcześniej algorytmie przypadek braku połączenia lub nawiązanie nieprawidłowego połączenia.

W poprzednim algorytmie w przypadku uzyskania nieprawidłowego połączenia bądź jego braku, przechodziliśmy do ostatniego kroku, w którym odkładaliśmy słuchawkę. Kończyło się Działanie algorytmu. W tej sytuacji powinniśmy rozpocząć raz jeszcze jego wykonywanie, nie zostało to jednak opisane w konstrukcji.

Rozbudujemy teraz algorytm, tak by powtarzano wybieranie numeru aż do uzyskania połączenia. Dopiszemy w tym celu polecenie będące drugim rodzajem instrukcji iteracyjnej:

Powtarzaj wykonywanie instrukcji A aż do spełnienia warunku W.

Czym jest instrukcja A, czym warunek W ?

Instrukcja A – podniesienie słuchawki, wybranie numeru

Warunek W – uzyskanie połączenia z wybranym numerem

Algorytm (wersja 4):

1. Czy słuchawka jest odłożona ? Jeśli tak, to przejdź do kroku 2. Jeśli nie, to odłóż słuchawkę.
2. Podnieś słuchawkę.
3. Wybierz cyfrę 6.
4. Wybierz cyfrę 1.
5. Wybierz cyfrę 6.
6. Wykonaj czynność cztery razy: Wybierz cyfrę 2.
7. Czy połączyłeś się z koleżanką ? Jeśli tak, to przejdź do kroku 8. Jeśli nie, to przejdź do kroku 9.
8. Zaproś koleżankę.
9. Odłóż słuchawkę.

W algorytmie telefonicznego zaproszenia koleżanki dodaliśmy instrukcję, która uwzględniła, że po nieudanej próbie uzyskanie połączenia należy odłożyć słuchawkę, a dopiero później ponownie ją podnieść i spróbować raz jeszcze. W przypadku pierwszej próby nawiązania połączenia również sprawdzamy odłożenie słuchawki.

W naszym algorytmie sprawdzamy warunek uzyskania połączenia z koleżanką (krok 4). Czasami zdarza się jednak, że już po podniesieniu słuchawki słychać, że linia jest zajęta. Należałoby wtedy odłożyć słuchawkę i ponownie ją podnieść. Jeśli okazałoby się, że nadal słychać w niej sygnał zajętości linii czynność należałoby powtórzyć. Musielibyśmy wykonywać te czynności dopóki linia nie byłaby „czysta”. W takim przypadku stosujemy instrukcję, która działa według schematu:

Dopóki warunek W jest spełniony, wykonuj instrukcję A.

Algorytm (w wersji 5) wygląda tak :

1. Czy słuchawka jest odłożona ? Jeśli tak, to przejdź do kroku 2. Jeśli nie, to odłóż słuchawkę.
2. Podnieś słuchawkę.
3. Czy linia jest zajęta ? Jeśli Tak, to: Odłóż słuchawkę. Podnieś słuchawkę. Przejdź do kroku 3. Jeśli Nie, to przejdź do kroku 4.
4. Wybierz cyfrę 6.
5. Wybierz cyfrę 1.
6. Wybierz cyfrę 6.
7. Wykonaj czynność cztery razy: Wybierz cyfrę 2.
8. Czy połączyłeś się z koleżanką ? Jeśli tak, to przejdź do kroku 9. Jeśli nie, to przejdź do kroku 1.
9. Zaproś koleżankę.
10. Odłóż słuchawkę.

Jak widzimy w kroku 3 występuje pętla, w której sprawdzamy najpierw warunek, a dopiero potem wykonywana jest instrukcja opisana przez kroki a, b. Jeśli warunek nie jest spełniony, to instrukcja nie zostanie wykonana ani razu.

Algorytm cykliczny (algorytm z pętlą, algorytm iteracyjny) został tutaj przedstawiony w postaci listy kroków. Algorytmy mogą być przedstawiane również w postaci graficznej zwanej schematem blokowym.

Zadanie 3. Napisz ten sam algorytm (w wersji 5) za pomocą schematu blokowego

ALGORYTMY REKURENCYJNE (algorytm rekurencyjny)

W informatyce możemy realizować również szczególny rodzaj powtórzeń bez konieczności stosowania pętli. Jest to technika **rekurencji**. Na razie nie będziemy się nią tutaj zajmować.