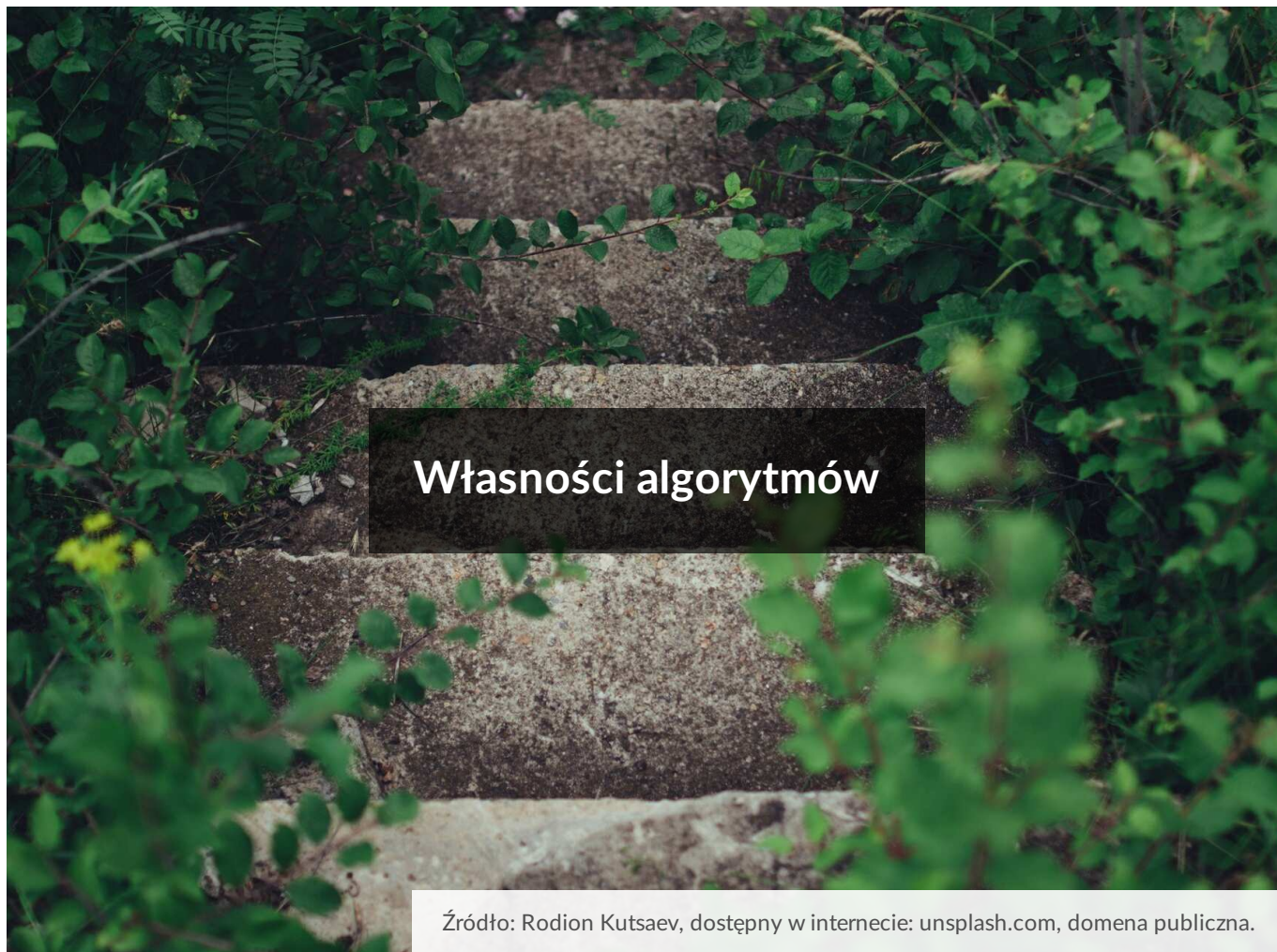




Własności algorytmów

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Audiobook](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

W tym e-materiale poznamy własności oraz cechy algorytmu.

Więcej informacji o algorytmach znajdziesz w e-materiałach:

- [Algorytmika](#),
- [Algorytmy na co dzień](#),
- [Specyfikacja problemu algorytmicznego](#).

Więcej zadań z zakresu algorytmiki? Zajrzyj do [Algorytmika – zadania maturalne](#).

Twoje cele

- Wyjaśnisz, kiedy można nazwać algorytm poprawnym, a kiedy częściowo poprawnym.
- Scharakteryzujesz własności algorytmów.
- Przeanalizujesz, jak rozpoznawać algorytmy poprawne.

Przeczytaj

Poznaliśmy już definicję algorytmów i ich przykładowe zapisy.

Algorytm

Algorytmem nazywamy przepis postępowania, który prowadzi do rozwiązania ustalonego problemu. Przepis ten określa ciąg czynności elementarnych, które należy w tym celu wykonać.

Ważne!

W wielu książkach poświęconych algorytmice wykorzystuje się przykład przepisu kulinarnego, by wyjaśnić najważniejsze definicje związane z algorytmami. W swojej książce *Piramidy, szyszki i inne konstrukcje algorytmiczne* profesor Maciej Sysło wykazuje, że książka kucharska raczej nie powinna być traktowana jako zbiór algorytmów.

Jednoznaczność algorytmu

Każdy algorytm, musi być **jasno określonym ciągiem poleceń**. Jeśli polecenia są jasno zdefiniowane, to powinny być również **jednoznaczne**.

To właśnie jednoznaczność jest jedną z cech każdego algorytmu. Cecha ta nazywana jest również **określonością**.

Jednoznaczność algorytmu gwarantuje, że dla tych samych danych wejściowych otrzymamy taki sam wynik. Warto zaznaczyć, że istnieją algorytmy, które w różnym stopniu realizują **optymalność**, w zakresie swojej implementacji. Przez co wyniki poszczególnych implementacji mogą się od siebie różnić. Jednak prawidłowa implementacja algorytmu zwróci poprawny wynik.

Skończoność algorytmu

Lista kroków musi w każdym algorytmie doprowadzać do jego zakończenia i powinna się wykonać w **skończonej** liczbie kroków (własność stopu).

Poprawność algorytmu

Algorytm jest całkowicie poprawny względem warunków początkowego i końcowego, gdy dla wszystkich danych zgodnych ze specyfikacją obliczenia algorytmu kończą się i wyniki spełniają warunek końcowy.

Przykładowo, chcemy ugotować zupę (np. krupnik) według przepisu (algorytmu). Przygotowujemy niezbędne składniki i wykonując kolejne kroki zgodnie z przepisem, otrzymujemy krupnik. Problem powstałby w sytuacji, gdybyśmy dysponując tym przepisem, chcieli ugotować rosół, a nie krupnik. Nie udalibyśmy się przecież do twórcy przepisu na krupnik, twierdząc, że jego przepis jest błędny.

A zatem o ile rosół w przypadku przepisu byłby poprawnym wynikiem, to nie byłby wynikiem przez nas oczekiwanym, co jednak wcale nie skreśla tego przepisu jako poprawnego.

Pamiętajmy więc, że nawet jeśli algorytm będzie poprawny, może z jakiegoś powodu nie spełnić naszych oczekiwań – choćby właśnie dlatego, że użyjemy nie tego algorytmu, którego potrzebowaliśmy.

Adekwatność algorytmu

Jednakże, gdyby ktoś określił przepis na rosół jako przepis na krupnik, to algorytm ten byłby błędny, ponieważ nie realizowałby założonego przez nas celu.

Własność ta nazywana bywa również **adekwatnością** – właśnie ze względu na to, że wynik powinien być zgodny z przyjętym celem.

Innym powodem, dla którego powyższą własność nazywa się czasem adekwatnością, jest to, że aby algorytm był poprawny, musi spełniać konkretne warunki. Dlatego też powiedzenie, że „algorytm nie jest poprawny, ponieważ nie spełnia własności poprawności”, może się wydawać nie do końca intuicyjne i być uznane za pleonazm. Musimy to wyraźnie podkreślić: **własność poprawności a poprawność algorytmu to nie jest to samo**. Warto o tym pamiętać, ponieważ istnieją pojęcia związane z poprawnością, częściową poprawnością oraz niepoprawnością algorytmu, które będziemy omawiać w dalszej części lekcji.

Złożoność algorytmu

Bardzo ważną cechą algorytmu jest jego **złożoność**, czasami nazywana też **efektywnością**. Każdy algorytm ma pewną złożoność obliczeniową i pamięciową.

Złożoność pamięciowa oznacza, ile komórek pamięci zajmie dany algorytm. Z racji tego, że algorytmy są uniwersalne, zwraca się uwagę jedynie na to, ile komórek pamięci zajmują elementy zdefiniowane przez algorytm, bez uwzględnienia tego, co dodatkowo wykorzystuje sam język.

Złożoność obliczeniowa liczona jest na podstawie liczby operacji dominujących, wykonywanych w trakcie tego algorytmu. Operacja dominująca to taka, która jest

realizowana w algorytmie najczęściej. Nie śledzi się więc wszystkich operacji, a jedynie te dominujące.

W przypadku algorytmów można podawać trzy różne złożoności:

- **pesymistyczne,**
- **średnie,**
- **optymistyczne.**

Do zapisywania złożoności używa się notacji wielkiego O, omegi Ω lub thety Θ .

Słownik

drzewo

struktura danych w której możemy wyróżnić takie elementy, jak: wierzchołki, krawędzie, liście i korzeń

kompresja

zmiana zapisu informacji polegająca na zmniejszeniu redundancji (nadmiarowości), dzięki czemu zmniejsza się cały rozmiar pliku

optymalność

najlepsza możliwa opcja, zachodząca w określonych warunkach

Audiobook

Polecenie 1

Zapoznaj się z audiobookiem i wykonaj ćwiczenie.

Audiobook można wysłuchać pod adresem: <https://zpe.gov.pl/b/PuFD5ITmN>

Czy każdy algorytm jest poprawny?

Algorytmy są w naszym życiu czymś powszechnym i wszechobecnym. Zazwyczaj kierujemy się nimi nieświadomie, choćby udzielając pierwszej pomocy zgodnie z medyczną procedurą. Pamiętajmy, że aby algorytm można było nazwać algorytmem poprawnym, musi on być jednoznaczny, skończony oraz spełniać własność poprawności (lub inaczej adekwatności).

Algorytm częściowo poprawny

Istnieje jednak również pojęcie algorytmu częściowo poprawnego. Charakteryzuje się on tym, że dla poprawnych danych zwróci poprawny wynik, o ile się zatrzyma. Znaczy to tyle, że algorytm ten nie ma własności stopu – czyli może nie być skończony.

Algorytmy częściowo poprawne mogą być stosowane w praktyce. Możemy na przykład powiedzieć, że liczb całkowitych jest nieskończenie wiele, ale są one przeliczalne – zatem teoretycznie jesteśmy w stanie powiedzieć, ile ich jest. Algorytm, który miałby obliczyć, ile jest liczb całkowitych, działałby w nieskończoność, gdyż samych liczb jest nieskończenie wiele, ale gdyby hipotetycznie kiedyś się zatrzymał (pomijamy tutaj kwestie sprzętowe), to zwróciłby nam poprawną moc zbioru liczb całkowitych. Jest to przykład stosunkowo abstrakcyjny, ale dobrze ilustruje zasadę, która mówi, że algorytm, jeśli się wykona w całości, zwróci poprawny wynik.

Innym przykładem algorytmu częściowo poprawnego jest szukanie największej liczby z przedziału liczb całkowitych podzielnej przez wszystkie swoje cyfry (nie licząc zera). Gdyby taka liczba została znaleziona, wynik byłby poprawny, ale to, czy kiedykolwiek zostanie znaleziona, może już być kwestią kłopotliwą, ponieważ liczb całkowitych jest – jak wiemy – nieskończenie wiele. Dla liczby 1 wynikiem jest oczywiście prawda, podobnie jak dla każdej liczby dodatniej mniejszej niż 10, jednak dla większych liczb problem ten jest bardziej złożony.

Algorytm niepoprawny

Istnieją również algorytmy niepoprawne. To takie algorytmy, które dla poprawnych danych wejściowych mogą się nigdy nie zatrzymać, a jeśli się już zatrzymają, mogą dać wynik niepoprawny.

Nie należy jednak mylić tego pojęcia z algorytmem niestabilnym. Algorytmy niestabilne zawierają błędy zaokrągleń wynikające po prostu z ograniczeń sprzętowych komputerów. Natomiast algorytmy niepoprawne mogą dawać kompletnie złe wyniki, niemające nic wspólnego z wynikiem oczekiwanym.

Algorytmów niepoprawnych nie należy również mylić z algorytmami zachłannymi, które z założenia szukają rozwiązań lokalnie optymalnych. Wynikiem algorytmu zachłannego nie jest więc nigdy z założenia wynik optymalny, zatem nie można tu powiedzieć, że wynik jest niepoprawny, ponieważ on z założenia nie ma być idealny.

Samo podanie przykładu algorytmu niepoprawnego nie jest proste. Załóżmy, że zabieg medyczny to pewna forma algorytmu. O ile każdy zabieg się skończy, to nawet jeśli zostanie przeprowadzony podręcznikowo, nie daje gwarancji na pozytywne zakończenie.

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Ćwiczenie 1

Wskaż, inne niż podane w audiobooku, przykłady algorytmów niepoprawnych i częściowo poprawnych.

Polecenie 2

Zastanów się, dlaczego przepis kulinarny niekoniecznie musi być algorytmem. Odpowiedź uzasadnij.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Dokończ zdanie.

Własność stopu to inaczej...

- ☐ adekwatność.
- ☐ własność znaku.
- ☐ efektywność.
- ☐ skończoność.

Ćwiczenie 2



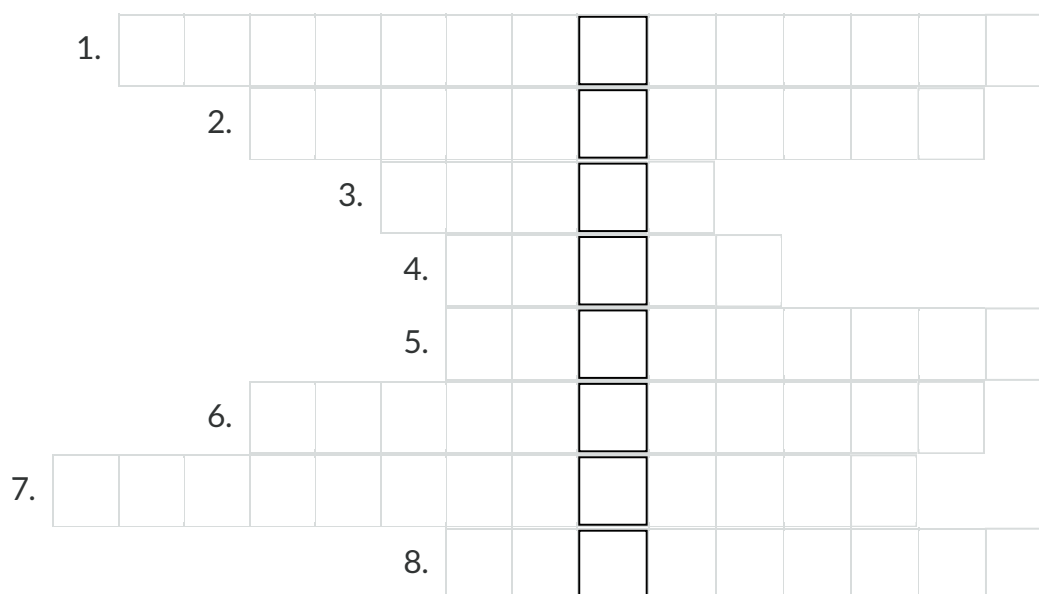
Wskaż wszystkie cechy algorytmu poprawnego.

- ☐ skończoność
- ☐ efektywność
- ☐ adekwatność
- ☐ przekształcalność
- ☐ jednoznaczność

Ćwiczenie 3



Rozwiąż krzyżówkę. Słowa zapisuj małymi literami.



1. Cecha każdego algorytmu.
2. Inna nazwa na jednoznaczność.
3. Jedna z notacji używana do określania złożoności, litera greckiego alfabetu.
4. Jedna z własności algorytmu.
5. ... drzewiasta
6. Inaczej złożoność.
7. Przeniesienie pseudokodu do formy programu to
8. Zmiana zapisu informacji polegająca na zmniejszeniu redundancji informacji.

Ćwiczenie 4



Zaznacz każdy opis odpowiednim kolorem.

☐ algorytm poprawny ☐ algorytm niepoprawny ☐ algorytm częściowo poprawny

Algorytm, który w skończonej liczbie kroków zwraca poprawny wynik.

Algorytm, który może się nigdy nie zatrzymać, ale jeśli się zatrzyma, to zwróci poprawny wynik.

Algorytm, który może się nigdy nie zatrzymać, ale jeśli się zatrzyma, to i tak nie ma gwarancji na zwrócenie poprawnego wyniku.

Ćwiczenie 5



Dokończ zdanie.

Własność jednoznaczności to inaczej...

☐ złożoność.

☐ adekwatność.

☐ skończoność.

☐ określoność.

☐ własność poprawności.

Ćwiczenie 6



Uzupełnij tekst właściwymi elementami.

Algorytm możemy zdefiniować jako jasno określony , który w czasie doprowadzi nas do jakiegoś problemu lub wykonania jakiegoś zadania.

zbiór poleceń

nieskończonym

skończonym

rozwiązania

ciąg poleceń

rozwiązania

Ćwiczenie 7



Wskaż poprawne rodzaje oceny złożoności.

☐ optymistyczna

☐ adekwatna

☐ optymalna

☐ efektywna

☐ średnia

☐ warunkowa

☐ pesymistyczna

Ćwiczenie 8



Wskaż własność wspólną dla wszystkich algorytmów (poprawny, niepoprawny, częściowo poprawny).

☐ skończoność

☐ określoność

☐ jednoznaczność

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Własności algorytmów

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania).

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

1) w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą;

Treści nauczania – wymagania szczegółowe

Klasy IV–VI

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2) formułuje i zapisuje w postaci algorytmów polecenia składające się na:

- a) rozwiązywanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
- b) osiągnięcie postawionego celu, w tym znalezienie elementu w zbiorze nieuporządkowanym lub uporządkowanym, znalezienie elementu najmniejszego i największego,

3) w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

1) projektuje, tworzy i zapisuje w wizualnym języku programowania:

- a) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,

2) testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów;

Klasy VII i VIII

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

1) formułuje problem w postaci specyfikacji (czyli opisuje dane i wyniki) i wyróżnia kroki w algorytmicznym rozwiązywaniu problemów. Stosuje różne sposoby przedstawiania algorytmów, w tym w języku naturalnym, w postaci schematów blokowych, listy kroków;

2) stosuje przy rozwiązywaniu problemów podstawowe algorytmy:

- a) na liczbach naturalnych: bada podzielność liczb, wyodrębnia cyfry danej liczby, przedstawia działanie algorytmu Euklidesa w obu wersjach iteracyjnych (z odejmowaniem i z resztą z dzielenia),

3) przedstawia sposoby reprezentowania w komputerze wartości logicznych, liczb naturalnych (system binarny), znaków (kody ASCII) i tekstów;

4) rozwija znajomość algorytmów i wykonuje eksperymenty z algorytmami, korzystając z pomocy dydaktycznych lub dostępnego oprogramowania do demonstracji działania algorytmów;

5) prezentuje przykłady zastosowań informatyki w innych dziedzinach, w zakresie pojęć, obiektów oraz algorytmów.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

1) projektuje, tworzy i testuje programy w procesie rozwiązywania problemów. W programach stosuje: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje oraz zmienne i tablice. W szczególności programuje algorytmy z działu I pkt 2;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Wyjaśnisz, kiedy można nazwać algorytm poprawnym, a kiedy częściowo poprawnym.
- Scharakteryzujesz własności algorytmów.
- Przeanalizujesz, jak rozpoznawać algorytmy poprawne.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- metody aktywizujące.

Formy pracy:

- praca indywidualna;
- praca w parach;

- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Własności algorytmów”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla uczniom temat zajęć oraz cele. Prosi, by na ich podstawie uczniowie sformułowali kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Jeżeli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi o indywidualne zapoznanie się z treścią zawartą w sekcji „Przeczytaj”. Każdy uczestnik zajęć podczas cichego czytania wynotowuje najważniejsze kwestie poruszane w tekście.
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Audiobook”. Uczniowie wspólnie zapoznają się z treścią zawartego w niej multimedium. Następnie nauczyciel czyta treść ćwiczenia nr 1 („Wskaż inne niż podane w audiobooku przykłady algorytmów niepoprawnych i częściowo poprawnych”) oraz prosi uczniów, aby wykonali je w parach.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenia nr 1-8, a następnie dzielą się wynikami swojej pracy z kolegą lub koleżanką.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się”.

Praca domowa:

1. Uczniowie wykonują polecenie: „Podaj przykłady algorytmów całkowicie poprawnych i częściowo poprawnych”.

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj”, „Audiobook”, „Sprawdź się” jako materiał do lekcji powtórkowej.