

Projektowanie i analiza algorytmów

Przedmiotowy system oceniania dla klasy 2f

I. Główne założenia PSO

Ocenianie uczniów na lekcjach informatyki:

- spełnia założenia wewnątrzszkolnego systemu oceniania;
- powinny być systematyczne;
- powinny pełnić rolę motywującą.

Przy ocenianiu należy uwzględnić:

- umiejętność pozyskiwania i wykorzystywania zdobytej wiedzy,
- aktywność, zaangażowanie i inicjatywę,
- kreatywność i odwagę twórczego myślenia,
- umiejętność współpracy i atmosferę panującą w grupie,
- samodzielność w formułowaniu poglądów i w podejmowaniu decyzji,
- wytrwałość i odpowiedzialność.

II. Obszary aktywności podlegające ocenie

Mierzeniu osiągnięć uczniów powinny służyć następujące aktywności:

- przygotowanie do lekcji (zadania domowe),
- ćwiczenia wykonywane na lekcji,
- zaangażowanie w realizację projektów grupowych,
- krótkie prace kontrolne (kartkówki),
- sprawdziany,
- udział w konkursach.

Uczniowie są zobowiązani do aktywnej i regularnej pracy poza zajęciami stacjonarnymi w systemie e-learningowym (<http://e.vlo.poznan.pl/>), który umożliwia zastosowanie elementów tzw. kształcenia wyprzedzającego i stanowi miejsce kontrolowanej współpracy uczniów i komunikacji z nauczycielem.

III. Wymagania na poszczególne oceny

Wymagania na poszczególne oceny podzielono na cztery grupy:

Na ocenę dopuszczającą i dostateczną (wytłuszczono)	Na ocenę dobrą i bardzo dobrą (wytłuszczono)
1. Elementy matematyki dyskretnej	
Uczeń:	
<ul style="list-style-type: none"> – rozumie pojęcie logarytmu – rozumie pojęcie drzewa binarnego – wyjaśnia związek logarytmu (o podstawie 2) z wysokością drzewa binarnego – rozumie pojęcie niezmiennika – i stosuje to pojęcie w praktyce – rozumie pojęcia: zbiór i podzbiór, ciąg i podciąg – rozróżnia wzór ogólny (zwały) ciągu od wzoru rekurencyjnego – rozumie pojęcie sumy ciągu liczbowego – oblicza sumę wyrazów ciągu arytmetycznego – zna podstawowe pojęcia kombinatoryczne (permutacja, kombinacja) 	<ul style="list-style-type: none"> – poprawnie stosuje symbole silni – i symbolu Newtona – posługuje się podstawowymi zasadami kombinatorycznymi (zasada dodawania, zasada mnożenia, zasada szufladkowa Dirichleta) – posługuje się zaokrągleniami całkowitymi (funkcjami $\lfloor \cdot \rfloor$ i $\lceil \cdot \rceil$) – potrafi szacować, – rozumie pojęcie rzędu wielkości – rozumie pojęcia logiczne: alternatywa, koniunkcja, implikacja, równoważność – rozumie pojęcie dowodu (warunek konieczny, warunek wystarczający)

2. Techniki projektowania algorytmów

Uczeń:

<p>– wyjaśnia pojęcie algorytmu i podaje proste przykłady</p> <p>– potrafi analizować pseudokod algorytmu</p> <p>– wyjaśnia znaczenie faz algorytmizacji</p> <p>– stosuje metodę zstępującą i metodę wstępującą przy rozwiązywaniu problemów algorytmicznych</p> <p>– rozumie istotę iteracji w rozwiązywaniu problemów algorytmicznych</p> <p>i wyjaśnia jej działanie na przykładzie (m.in. algorytm Euklidesa)</p> <p>– zna ogólne techniki projektowania algorytmów:</p> <ul style="list-style-type: none">– wyczerpującego przeszukiwania (ang. brute force)– dziel i zwyciężaj (ang. divide-and-conquer) <p>i potrafi zilustrować je przykładami problemów algorytmicznych, np.:</p> <ul style="list-style-type: none">– sortowanie przez wybór i sortowanie bąbelkowe– przeszukiwanie liniowe (sekwencyjne)– przeszukiwanie binarne (logarytmiczne)– obliczanie wartości wielomianu (schemat Hornera) <p>– zna techniki projektowania algorytmów optymalizacyjnych:</p> <ul style="list-style-type: none">– podejście zachłanne– przeszukiwanie z nawrotami <p>i potrafi zilustrować je przykładami problemów algorytmicznych, np.:</p> <ul style="list-style-type: none">– zachłanne wydawanie reszty– problem konomana– wyjście z labiryntu– rozmieszczenie hetmanów na szachownicy <p>– stosuje różne strategie algorytmiczne podczas projektowania algorytmów ad hoc dla prostych sytuacji problemowych</p> <ul style="list-style-type: none">– zapisuje zaprojektowane algorytmy je w postaci listy kroków lub pseudokodu	<p>– zna ogólne techniki projektowania algorytmów:</p> <ul style="list-style-type: none">– zmniejsz i zwyciężaj (ang. decrease-and-conquer)– przekształć i zwyciężaj (ang. transform-and-conquer) <p>i potrafi zilustrować je przykładami problemów algorytmicznych, np.:</p> <ul style="list-style-type: none">– szybkie potęgowanie– szukanie wzorca w tekście– problem komiwojażera,– problem plecakowy– generowanie podzbiorów– generowanie permutacji <p>– rozumie istotę rekurencji w rozwiązywaniu problemów algorytmicznych</p> <p>i wyjaśnia jej działanie na przykładzie (m.in. algorytm Euklidesa)</p> <p>– zna techniki projektowania algorytmów optymalizacyjnych:</p> <ul style="list-style-type: none">– podejście zachłanne– programowanie dynamiczne <p>i potrafi zilustrować je przykładami problemów algorytmicznych, np.:</p> <ul style="list-style-type: none">– minimalizacja łączenia par– minimalne drzewo rozpinające– najkrótsza ścieżka w grafie– wyjście z labiryntu– pakowanie plecaka <p>– prowadzi rozumowanie celem uzasadnienia, że strategia prowadzi do znalezienia optymalnego (najlepszego z możliwych) rozwiązania</p> <p>– zna typy innych ważnych problemów algorytmicznych, np. problemy geometryczne, numeryczne, kombinatoryczne (decyzyjne i optymalizacyjne), teoriografowe, teorioliczbowe i tekstowe (na tekstach)</p> <p>– rozumie zasadnicze znaczenie algorytmów przeszukiwania i algorytmów sortowania w dziedzinie algorytmicznego rozwiązywania problemów</p>
---	--

2. 3. Metody analizy algorytmów

Uczeń:

<p>– analizuje krok po kroku efekt działania algorytmu obliczeniowego (numerycznego) zapisanego w postaci pseudokodu lub schematu blokowego</p> <p>– prowadzi rozumowanie indukcyjne (uogólnia dostrzeżone prawidłowości)</p> <p>– analizuje poprawność (zgodność ze specyfikacją) algorytmu zapisanego w postaci pseudokodu lub schematu blokowego</p> <p>– rozumie pojęcie rozmiaru danych wejściowych</p> <p>– rozumie pojęcie operacji podstawowej (najbardziej kosztownej) algorytmu</p> <p>– rozumie potrzebę stosowania algorytmów przybliżonych oraz obliczeń równoległych</p> <p>– opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego i efektywnego rozwiązania wybranego problemu algorytmicznego</p> <p>– poprawnie sumuje wyrazy ciągów liczbowych</p>	<p>– dowodzi poprawności prostych algorytmów, z użyciem pojęcia niezmiennika</p> <p>– opisuje własności algorytmów na podstawie ich analizy</p> <p>– oblicza liczbę operacji podstawowych wykonywanych przez algorytm</p> <p>– wyjaśnia pojęcie efektywności (pracochłonności) algorytmu, porównując działanie dwóch algorytmów rozwiązującej ten sam problem</p> <p>– rozumie symbol oszacowania asymptotycznego $O(-)$</p> <p>– szacuje pesymistyczną i oczekiwaną (średnią) złożoność obliczeniową wybranych prostych algorytmów oraz złożoność pamięciową algorytmu,</p> <p>– poprawnie stosuje symbole silni i symbolu Newtona, funkcji $\lfloor \rfloor$ i $\lceil \rceil$</p> <p>– rozumie pojęcie optymalności algorytmu</p> <p>– rozumie ograniczenia algorytmiki, wyjaśnia na przykładzie (np. problem ko-miwojażera), na czym polega problem złożony obliczeniowo oraz zjawisko eksplozji kombinatorycznej</p>
--	---

IV. Sprawdzanie i ocenianie osiągnięć uczniów

Poniżej znajdują się informacje o sprawdzaniu i ocenianiu osiągnięć (oraz zasadach ustalania ocen semestralnej i końcowej).

rodzaj aktywności	waga	komentarz
sprawdzian i poprawa sprawdzianu	3	Pracę napisaną na ocenę niedostateczną należy pisać ponownie.
kartkówka	1	Dotyczy zagadnień omawianych podczas trzech ostatnich jednostek lekcyjnych.
zadanie domowe	1	
projekt przedmiotowy	2	
szczególna aktywność	1	
szczególne osiągnięcia (np. w konkursie)	2, 3, 6	za udział w II etapie Olimpiady Informatycznej – ocena wagi 3 za udział w III etapie Olimpiady Informatycznej – ocena wagi 6

W przypadku prac pisemnych obowiązują następujące progi procentowe dla ocen:

<98%; 100%> cel

<95%; 98%) bdb+

<90%; 95%) bdb

<85%; 90%) db+
<75%; 85%) db
<70%; 75%) dst+
<60%; 70%) dst
<55%; 60%) dop+
<45%; 55%) dop
<0%; 45%) ndst

Na podstawie średniej ważonej ocen cząstkowych (odpowiednio z I semestru i całego roku) ustalane są oceny: semestralna i końcowa:

od 5,6 – celująca
od 4,6 – bardzo dobra
od 3,6 – dobra
od 2,6 – dostateczna
od 1,6 – dopuszczająca
poniżej 1,6 ndst

V. Sposób informowania o postępach i ich braku

Informacje o postępach uczniów zamieszczane są w szkolnym dzienniku.

VI. Zasady poprawiania ocen

Sprawdzian napisany na ocenę niedostateczną uczeń jest zobowiązany napisać ponownie w ciągu dwóch tygodni od otrzymania oceny (wpisania oceny do dziennika). Obie oceny są wpisywane do dziennika.

W przypadku innych niż niedostateczna ocen ze sprawdzianu uczeń może pisać pracę, jeśli jest zainteresowany poprawą oceny.

Poprawa odbywa się w czasie dyżuru nauczyciela (tzw. 19h).

VII. Zasady wglądu uczniów i rodziców (opiekunów) do prac pisemnych

Prace kontrolne są oddawane (lub udostępniane w systemie internetowym) uczniom, którzy są zobowiązani zachować je do końca roku szkolnego.

Sprawdziany w formie papierowej są udostępniane do wglądu uczniom i rodzicom – po upływie co najwyżej dwóch tygodni praca powinna zostać oddana nauczycielowi.

Nauczyciel ma obowiązek przechowywać sprawdziany do końca roku szkolnego.

Kopia sprawdzianu w wersji elektronicznej jest dostępna do końca roku szkolnego.

Paweł Perekietka
pperekietka@vlo.poznan.pl