

**Program nauczania informatyki dla zajęć pozalekcyjnych
dla uczniów uzdolnionych informatycznie
na I rok nauczania**

Przewidziany na 15 spotkań po 4 godziny lekcyjne (razem 60 godzin lekcyjnych)

Autor – Małgorzata Piekarska
VI Liceum Ogólnokształcące w Bydgoszczy

1. Informacje ogólne

Przedstawiony program powstał w celu podzielenia się doświadczeniami z pracy z uczniem uzdolnionym w zakresie algorytmiki i programowania. Jest on skierowany do nauczycieli informatyki w szkołach ponadpodstawowych, którzy chcieliby pomóc swoim zdolnym uczniom rozwinąć informatyczne skrzydła, skierować ich na drogę, której jednym z etapów jest udział w Olimpiadzie Informatycznej.

Swoim doświadczeniem dzieli się tu nauczyciel, z wieloletnim stażem pracy z uczniem zdolnym, nauczyciel informatyki w VI Liceum Ogólnokształcącym w Bydgoszczy, którego uczniowie każdego roku biorą udział w Olimpiadzie Informatycznej. Przedstawiony program jest wspólnym wykładnikiem wieloletniej pracy w szkole, obserwacji i rozmów z uczniami, a także szeroko rozumianej współpracy i wymiany doświadczeń z innymi nauczycielami-opiekunami olimpijczyków z wielu ośrodków w Polsce. Program przewidziany jest na pierwszy rok nauczania.

2. Cele.

- Celem, dla którego powstał ten program jest próba udzielenia wsparcia zdolnym uczniom ze szkół ponadpodstawowych i ponadgimnazjalnych oraz ich nauczycielom w rozplanowaniu wspólnej pracy na zajęciach pozalekcyjnych w pierwszym roku nauki.
- Celem nadrzędnym programu, przy spełnieniu wszystkich przyjętych założeń, będzie zwiększenie motywacji do dalszego rozwoju, potwierdzone po pierwszym roku udziałem uczestników zajęć w pierwszym etapie Olimpiady Informatycznej.

3. Efekty kształcenia:

- Rozwój umiejętności programowania w C++,
- Poznanie metod i strategii projektowania algorytmów,
- Poznanie nowych struktur danych,

- Kształtowanie umiejętności twórczego rozwiązywania zadań z użyciem komputera, rozwój aktywności twórczej,
- Podniesienie wiary w siebie i świadomości swoich mocnych stron,
- Zainteresowanie algorytmiką i programowaniem,
- Zachęcanie uczniów do szerszego samodzielnego pogłębiania wiedzy z algorytmiki i programowania,
- Kształcenie umiejętności organizacji pracy, odpowiedzialności za własny rozwój,
- Podjęcie przez ucznia próby udziału w Olimpiadzie Informatycznej lub innych konkursach algorytmicznych.

4. Opis wymaganych kompetencji uczniów i propozycja sposobu weryfikacji ich posiadania

Jednym z głównych założeń jest specyfika grupy docelowej – tematy zajęć przeznaczone są dla uczniów uzdolnionych informatycznie. Uczeń zdolny, w takim rozumieniu, to ten, który rozpoznaje problem, próbuje rozwiązać go w sposób twórczy, logiczny, prawidłowy i często oryginalny z użyciem komputera. Proponowane metody weryfikowania zdolności:

- A. Czasem uzdolnienia potwierdzają wcześniejsze osiągnięcia ucznia: potwierdzone sukcesem uczestnictwo w konkursach matematycznych czy informatycznych (algorytmicznych), przykładowo: Olimpiada Informatyczna Juniorów (wcześniej Olimpiada Informatyczna Gimnazjalistów), Konkurs Informatyczny Bóbr, Konkursy Wojewódzkie z Informatyki lub Matematyki i tym podobne.
- B. Obserwacje poczynąń ucznia na zajęciach szkolnych z matematyki czy informatyki. Podczas zajęć lekcyjnych można spostrzec pewne symptomy uzdolnień:
 - Uczeń jest ciekawy, dopytuje o kwestie nie objęte przez program nauczania, jest dociekliwy,
 - Jest otwarty na nowe rozwiązania, baczny i czujny obserwator,
 - Posiada obszerną wiedzę, często wykraczającą poza program nauczania,
 - Jego pomysły na rozwiązanie problemu są nietypowe, ciekawe, oryginalne i twórcze,
- C. Test predyspozycji z wykorzystaniem łamigłówek, zadań logicznych z konkursów informatycznych lub olimpiad (np. Olimpiada Informatyczna Juniorów – pytania tury testowej rok szkolny 2019/2020, Konkurs Informatyczny Bóbr – wszystkie edycje).

5. Procedury osiągnięcia celów

A. Metody pracy z uczniami

Akceptowalne i pożądane są wszystkie formy kształcenia stwarzające warunki do:

- rozwoju zainteresowań, zdolności i uzdolnień,
- poszerzania wiedzy,
- doskonalenia umiejętności
- zwiększenia, podtrzymania motywacji uczestników zajęć.

Przedstawiona koncepcja zakłada, że obszernym fragmentem początkowej fazy większości zajęć powinien być konkurs algorytmiczny, w czasie którego uczniowie będą próbować **samodzielnie rozwiązywać problemy** o różnym stopniu trudności. Nie ma lepszej metody niż **zadania praktyczne**, czyli samodzielne **doświadczenie**. Zadania, które uczniowie będą próbować rozwiązać przy pomocy komputera muszą być pieczołowicie dobrane, a ich poziom trudności zróżnicowany. Konkurs powinien być zabawą i sprawiać radość odkrywania, dlatego należy bezwzględnie stworzyć warunki zabawy, wyeliminować jakikolwiek stres, dlatego z początku dozwalamy by młodzież na cele konkursu łączyła się w maksymalnie 2-osobowe **drużyny** (większa ilość osób w drużynie powoduje demotywację i słabszy przyrost umiejętności, szczególnie niepożądane w początkowej fazie rozwoju). Przeprowadzanie konkursów można oprzeć o jedną z ogólnodostępnych platform z zadaniami (np. szkopol.edu.pl). W czasie niektórych zajęć (na przykład przy wprowadzaniu rekurencji) używać można testów i kwestionariuszy.

Na każdym zajęciach po zakończonym konkursie, znajduje się miejsce na **dyskusję przyjętych sposobów rozwiązań**, omówienie rozwiązań wzorcowych. Tu często pojawiają się **elementy wykładu** zagadnień poruszanych w zadaniach. Tu może rozpętać się prawdziwa **burza mózgow**. Nierzadko do omówienia zadań można poprosić jednego z uczniów, któremu udało się wykonać zadanie na maksymalną liczbę punktów. Stwarza to dodatkowy element aktywizujący i motywujący. Na zakończenie zajęć jest zawsze zadawana **praca domowa**, którą stanowią zadania związane z tematem zajęć. Praca domowa to również runda (tu koniecznie **indywidualna**) w konkursie na wymienionej platformie.

Punktacja uzyskana za zadania w ramach pojedynczych zajęć i w ramach prac domowych jest sumowana i tak stopniowo powstaje ogólna lista rankingowa uczestników (**grywalizacja** – zastosowanie mechaniki gier i współzawodnictwa w celu zwiększenia motywacji). Ponadto sukcesy w konkursach, jak i systematyczna praca w zadaniach domowych są nagradzane dodatkowymi ocenami z informatyki (za aktywność wykraczającą ponad program nauczania). Wysokie miejsca w rankingach mogą być również każdorazowo nagrodzone symboliczną, drobną nagrodą, np. cukierkiem, a oprócz tego można, za zgodą uczestników, publikować co tydzień w wybranym dniu listę rankingową np. na gazetce w szkolnej pracowni. Doświadczenie pokazuje, że element nagrody jest tu bardzo istotny i powinien być dostosowany do warunków i do specyfiki grupy.

Najlepsi uczniowie, którzy uzyskali w rankingu ogólnym, po wszystkich zajęciach i pracach domowych wysokie pozycje, mogą na przykład na koniec roku szkolnego przygotować w formie dodatkowego **projektu**, pod kierunkiem nauczyciela, konkurs na łamigłówki lub zadania logiczne dla wszystkich chętnych uczniów w szkole.

Dodatkową, nieodłączną formą kształcenia powinno być **samokształcenie**. Uczeń może z polecenia nauczyciela przygotować jakiś mini wykład, dowód itp. Poza tym dobrze jest stawiać uczniom jasne cele, rozwój dzielić na etapy, motywować do tworzenia swojego własnego planu rozwoju, a także do dokumentowania swoich poczynionych postępów, rozwiązanych zadań, **tworzenia swojego informatycznego portfolio**.

B. Środki dydaktyczne

Warunki, w tym infrastruktura, niezbędne do realizacji zajęć (sprzęt, oprogramowanie, zasoby internetowe).

- Komputer dla każdego uczestnika kółka, system operacyjny dowolny, konfiguracja dowolna, komputery powinny być wyposażone w kompilatory języka C++ (proponowane to: CodeBlocks, Dev C++ lub VS Code) i dostęp do Internetu, gdzie uczniowie zamiast używać kompilatora, mogą zamiennie pisać programy online (np. na platformie repl.it), a także by mogli w czasie konkursów wysyłać swoje rozwiązania na platformę z automatycznym sprawdzaniem (proponowany: szkopuł.edu.pl)
- Rzutnik, tablica z pisakami lub kredą ułatwiający prowadzenie wykładowej części zajęć,
- Dostęp dla nauczyciela do platformy z zadaniami (np. szkopuł.edu.pl)

6. Ewaluacja

Źródłem informacji o efektach pierwszego roku pracy będzie przede wszystkim wywiad - luźna rozmowa z uczniami, uczestnikami koła. Całoroczna obserwacja ich pracy i postępów w nauce powinna znaleźć potwierdzenie w ich systematycznie tworzonego portfolio. Poświadczeniem zdobytych umiejętności będzie podjęcie próby uczestnictwa w konkursach algorytmicznych, na przykład w Olimpiadzie Informatycznej, a także miejsce w rankingu uzyskane po całym roku zajęć.

7. Plan zajęć.

Sugerowany czas trwania każdych zajęć to **4 godziny lekcyjne (180 min)**. Wszystkie zajęcia mają podobny harmonogram:

- faza pierwsza - omówieniem zadań z prac domowych.
- faza wstępna – określenie tematu zajęć, celów. Krótki wstęp
- faza doświadczalna - konkurs z wykorzystaniem zaproponowanych zadań (jeśli przy zadaniu nie określono pochodzenia lub autora danego zadania, to znaczy że zadanie pochodzi z zasobów własnych)
- faza nauki / część wykładowa połączona z omówieniem zadań z bieżącego konkursu
- faza podsumowująca i utrwalająca – zadania do wykonania samodzielnego w domu (przekazanie uczniom że jest to integralna część nauki)

Lp.	Temat / cel zajęć	Treści programowe w odniesieniu do sylabusu	Przykładowe zadania
1	<p>Podstawy programowania w C++.</p> <p>Cele:</p> <ul style="list-style-type: none">– Opanowanie języka programowania w stopniu umożliwiającym zapis prostych algorytmów.– Zapoznanie z platformą szkopuł.edu.pl– Pierwsze algorytmy iteracyjne– Rola optymalizacji czasu pracy programu	<p>Środowisko programistyczne i jego obsługa</p> <p>Zmienne i ich typy, wyrażenia arytmetyczne</p> <p>Instrukcje strumieniowego wejścia i wyjścia, formatowanie wyjścia, synchronizacja wejścia i wyjścia strumieniowego</p> <p>Instrukcja warunkowa i operatory logiczne, typ logiczny.</p> <p>Zdania logiczne, złożone zdania logiczne.</p> <p>Instrukcja iteracji (for, while, do while)</p> <p>Typ napisowy i pętla przeglądająca kolejne znaki.</p> <p>Funkcje, parametry funkcji, zasięg zmiennych.</p> <p>Zapoznanie z systemem automatycznego sprawdzania rozwiązań uczniów, testowanie programów.</p>	<p>1.1.Trójkąt</p> <p>1.2.Prostokąt</p> <p>1.3Fizz Buzz</p> <p>1.4.Choinka</p> <p>1.5.Dzielniki</p> <p>1.6.Pierwszość liczby</p> <p>Zadanie domowe</p> <p>1.7.Dodawanie ułamków</p> <p>1.8.Uszkodzona klawiatura</p>

		<p>Czas działania programu, złożoność obliczeniowa (operacja wiodąca)</p> <p>Algorytmy: pierwiastkowy test pierwszości, wyznaczanie dzielników liczby, algorytm Euklidesa (wersja iteracyjna),</p>	<p>1.9.Urodziny</p> <p>1.10.Suma</p>
2	<p>Zabawa z tablicami jednowymiarowymi – część I</p> <p>Cele:</p> <ul style="list-style-type: none"> – Tablice jednowymiarowe jako podstawowy sposób na organizację danych w zadaniach olimpijskich – Poznanie podstawowych algorytmów i tricków do przetwarzania danych w tablicach 	<p>Tablice statyczne jednowymiarowe, deklaracja, przeszukiwanie liniowe, znajdowanie wartości najmniejszej /największej</p> <p>Zliczanie elementów</p> <p>Porządkowanie zbioru</p> <p>vector (STL)</p> <p>Biblioteka algorithm (funkcja sort)</p>	<p>2.1. Naj naj</p> <p>2.2.Figury (Jacek Tomaszewicz – ILOCAMP11)</p> <p>2.3.Monety (Jacek Tomaszewicz – ILOCAMP10)</p> <p>2.4. Zakład (Michał Majewski – ILOCAMP11)</p> <p>2.5. Permutacja (Jacek Tomaszewicz – ILOCAMP3)</p> <p>2.6. Odchudzanie (Jacek Tomaszewicz – ILOCAMP10)</p> <p>Zadanie domowe:</p> <p>2.7. Samochody (Jacek Tomaszewicz – ILOCAMP11)</p> <p>2.8.Pinezki (Joanna Bujnowska – ILOCAMP10)</p>
3	<p>Zabawa z tablicami jednowymiarowymi – część II</p> <p>Cele: j/w</p>	<p>Gąsienica</p> <p>Pojęcie metody siłowej - brute force.</p> <p>Złożoność algorytmów – notacja wielkiego O.</p> <p>Klasy funkcji złożoności (logarytmiczna, liniowa, $n \log n$, kwadratowa, sześcienna, wykładnicza).</p> <p>Podciąg o największej sumie</p>	<p>3.1. Dziewczynki (Jacek Tomaszewicz – ILOCAMP10)</p> <p>3.2. Najlepsze sumy (Matura 2005)</p> <p>3.3. Taśma (Jacek Tomaszewicz – ILOCAMP10)</p>

		<p>Sumy prefiksowe Lider zbioru</p>	<p>3.4. Bilet (Jacek Tomaszewicz – Podlaski turniej w programowaniu zespołowym)</p> <p>Zadanie domowe</p> <p>3.5. Długa taśma (Jacek Tomaszewicz – ILOCAMP4)</p> <p>3.6. Zliczacz liter (OiG1)</p> <p>2.11. Bierki (OIG1)</p> <p>2.12. Patyki (PA2002)</p>
4	<p>Podstawowe algorytmy na liczbach całkowitych.</p> <p>Cele:</p> <ul style="list-style-type: none"> – Poznanie niektórych zagadnień z teorii liczb – Zaznajomienie z rekurencją – Wykształcenie umiejętności rekurencyjnego postrzegania niektórych problemów 	<p>Funkcje rekurencyjne</p> <p>Algorytm Euklidesa w wersji rekurencyjnej</p> <p>NWW</p> <p>Sito Eratostenesa</p> <p>Rozkład na czynniki pierwsze</p>	<p>4.1. Zrozumieć rekurencję (OIJ)</p> <p>4.2. Korale (matura 2014)</p> <p>4.3. Rekurencja (OIG8)</p> <p>4.4. Wesoła małpka (Joanna Bujnowska – ILOCAMP11)</p> <p>4.5. Pierwsza czy złożona</p> <p>4.6. Rozkład na czynniki pierwsze</p> <p>Zadanie domowe</p> <p>4.7. Ilość dzielników liczby</p>
5	<p>Odwieczny problem z porządkiem.</p> <p>Cele:</p> <ul style="list-style-type: none"> – Poznanie niektórych metod sortowania danych. – Zademonstrowanie strategii Dziel i zwyciężaj 	<p>Sortowanie wybranymi metodami: przez wstawianie, przez scalanie, szybkie</p> <p>Dziel i zwyciężaj.</p> <p>Rekurencyjna realizacja funkcji sortujących.</p> <p>Funkcja sort (biblioteka algorithm)</p> <p>Złożoność czasowa funkcji sortujących.</p>	<p>5.1. Iloczyn – (Jacek Tomaszewicz- Podlaski Konkurs w Programowaniu Zespołowym)</p> <p>5.2. Karty (Codeforces)</p> <p>5.3. Programowanie zesp. (Codeforces)</p> <p>5.4. Sortowanie Biżuterii (OIG1)</p> <p>5.5. Izolator (OI9)</p>

			<p>Zadanie domowe</p> <p>5.6. Tory kolejowe (Jacek Tomaszewicz - V ILOCAMP)</p> <p>5.7. Flaga Polski (autor nieznany)</p>
6	<p>Porządek kluczem do szybkiego wyszukiwania. Wyszukiwanie binarne.</p> <p>Cele:</p> <ul style="list-style-type: none"> - Poznanie algorytmu wyszukiwania binarnego. - Poznanie kolejnego przykładu użycia strategii Dziel i zwyciężaj 	<p>Wyszukiwanie binarne (szukanie wartości).</p> <p>Złożoność logarytmiczna - dziel i zwyciężaj</p> <p>Wyszukiwanie binarne po wyniku.</p> <p>Biblioteka algorithm (lower_bound i upper_bound)</p> <p>Intuicyjne pojęcie logarytmu.</p>	<p>6.1. Wyszukiwanie binarne</p> <p>6.2. Zagadka Nicolò Tartaglii (kurs algorytmiki main2)</p> <p>6.3. Akcelerator (kurs algorytmiki main2)</p> <p>Zadanie domowe</p> <p>6.4. Analiza algorytmu (matura 2018)</p> <p>6.5. Pierwiastek</p> <p>6.6. Wartość równa indeksowi.</p>
7	<p>Tablice dwuwymiarowe.</p> <p>Cele:</p> <ul style="list-style-type: none"> - Biegłe posługiwanie się tablicami dwuwymiarowymi w języku C++, - Poznanie strategii przeszukiwania z nawrotami. - poznanie kolejnego przykładu zastosowania strategii Dziel i zwyciężaj. 	<p>Deklaracja tablicy dwuwymiarowej, podział na wiersze i kolumny.</p> <p>Sposób odwoływania się do elementów tablicy dwuwymiarowej.</p> <p>Problem 8 hetmanów, strategia przeszukiwania z nawrotami, rekurencja.</p>	<p>7.1. Sumy częściowe dwuwymiarowe</p> <p>7.2. Problem 8 hetmanów</p> <p>7.3. Zadanie Poziomki</p> <p>Zadanie domowe:</p> <p>7.4. Zadanie Macierz Jordana</p> <p>7.5. Zadanie Klatka Jordana</p>
8	<p>Systemy pozycyjne. Arytmetyka wielkich liczb.</p> <p>Szybkie potęgowanie modularne.</p> <p>Cele:</p> <ul style="list-style-type: none"> - Poznanie zadań olimpijskich w których liczby są zbyt wielkie na dowolny typ danych liczbowych w C++, 	<p>Duże liczby i ich sposoby reprezentacji w programie</p> <p>Dodawanie, odejmowanie, mnożenie i dzielenie całkowite przez dwa dużych liczb.</p> <p>Dodawanie i odejmowanie dużych liczb rzeczywistych.</p> <p>Systemy liczbowe (reprezentacja liczb w komputerze),</p> <p>Jak dużo miejsca zajmują liczby w systemie o podstawie p.</p>	<p>8.1. Julka (Krzysztof Diks PA2003)</p> <p>8.2. Sumujący Jaś (OIG1)</p> <p>Zadania domowe:</p> <p>8.3. Majątek Billa G. (Marcin Boniecki – pl.spoj.com)</p>

	<ul style="list-style-type: none"> – poznanie sposobów zamiany liczb między systemami pozycyjnymi, – poznanie algorytmu szybkiego potęgowania. – Szybkie potęgowanie modularne 	<p>Pojęcie logarytmu (ilość bitów w reprezentacji binarnej)</p> <p>Zamiana liczby z systemu dziesiętnego na dowolny i odwrotnie (przesunięcia bitowe).</p> <p>Jak szybko podnieść do potęgi przy dużym wykładniku.</p>	
9	<p>Algorytmy zachłanne i dynamiczne.</p> <p>Cele</p> <ul style="list-style-type: none"> – Poznanie kolejnych strategii rozwiązywania problemów. – Uświadomienie niekorzyści powstających przy korzystaniu z rekurencji w rozwiązywaniu niektórych problemów programistycznych. 	<p>Programowanie dynamiczne - konstruowanie optymalnego rozwiązania na podstawie wyników wcześniejszych obliczeń</p> <p>Metoda zachłanna – podejmowanie działań, które wydają się optymalne na daną chwilę.</p> <p>Algorytm wyznaczania liczb Fibonacciego</p> <p>Problem wydawania reszty</p>	<p>9.1. Oszczędności Bitka (Codeforces)</p> <p>9.2. Sklep Odzieżowy</p> <p>9.3. Liczby Fibonacciego</p> <p>Zadanie domowe:</p> <p>9.4. Klocki (OIG3)</p> <p>9.5. High Profit Only (z zasobów szkopol.edu.pl)</p> <p>9.6. Zachłanny waz atakuje (z zasobów szkopol.ed.pl)</p>
10	<p>Algorytmy tekstowe</p> <p>Cele</p> <ul style="list-style-type: none"> – Przedstawienie podstawowych pojęć związanych z przetwarzaniem napisów. – Algorytmy: wyszukiwania wzorca w tekście oraz test na palindrom. 	<p>Wymienia i rozumie podstawowe pojęcia dot. przetwarzania tekstów</p> <p>Przedstawia sposoby reprezentacji tekstów w programie w języku C++,</p> <p>Potrafi zastosować w typowych zadaniach wybrane algorytmy dla tekstów: Wyszukiwanie wzorca w tekście algorytmem naiwnym oraz algorytmem KMP,</p> <p>Zna algorytm sprawdzania czy podany tekst jest palindromem (alg.naiwny oraz algorytm Manachera)</p>	<p>10.1. Palindromy (Łukasz Jocz – ILOCAMP11)</p> <p>10.2. Palindroliczby (OIG3)</p> <p>Zadanie domowe:</p> <p>10.3. Szablon (OI12) – dla chętnych</p> <p>10.4. KMP (Rafał Nowak – pl.spoj.com)</p> <p>10.5. Antysymetria (OI17) – dla chętnych</p>
11	<p>Geometria</p>	<p>Stosuje technikę zmiatania</p> <p>Zna algorytmy: znalezienia tzw. wypukłej otoczki, znalezienia najbliższej pary punktów (dziel i zwyciężaj)</p>	<p>11.1. Przynależność punktu do odcinka</p> <p>11.2. Przycinanie się odcinków</p> <p>11.3. Punkt wewnątrz trójkąta</p>

	<p>Cele</p> <ul style="list-style-type: none"> – Poznanie sposobów rozwiązywania z pomocą komputera niektórych problemów z dziedziny geometrii analitycznej na płaszczyźnie. 	<p>najdalszej pary punktów (otoczka), sortowanie kątowe,</p> <p>Umie napisać program określający położenie punktu względem odcinka, wykryć fakt przecinania odcinków, zlokalizować położenie punktu względem wielokąta.</p>	<p>11.4.Punkt wewnątrz wielokąta wypukłego 11.5.Współliniowość punktów</p> <p>Zadanie domowe: 11.6.Wyspa (OI9)</p>
12	<p>Teoria grafów I</p> <p>Cele:</p> <ul style="list-style-type: none"> – Podstawy matematyki dyskretnej – grafy – poznanie podstawowych pojęć i algorytmów grafowych. – Poznanie jak i w jakim czasie za pomocą komputera możemy rozwiązywać niektóre problemy grafowe. 	<p>Uczeń poznaje podstawowe pojęcia teorii grafów: grafy nieskierowane i skierowane, wierzchołki, stopień wierzchołków, krawędzie, wagi krawędzi, ścieżki, cykle (Eulera i Hamiltona), spójność grafu, silna spójność, podgrafy, drzewa i lasy, grafy ważone, drzewa rozpinające, minimalne drzewa rozpinające, klasy grafów (dwudzielne, planarne), metody przechodzenia grafów, grafy specjalne</p> <p>Omawia i rozumie problem skojarzeń w grafach dwudzielnych</p> <p>Zna i potrafi określić złożoność a także napisać Algorytm Dijkstry – znajdujący najkrótszą drogę w grafach o nieujemnych wagach krawędzi.</p>	<p>12.1.Koneksje 12.2.Wszystkie żony 12.3.Drzewo i jego gałąź 12.4.Ilość możliwych przejazdów 12.5.Plansza (OIG8) 12.6. Dijkstra (Szkopul.edu.pl)</p> <p>Zadanie domowe: 12.7.Gildie (OI17)</p>
13	<p>Teoria grafów II</p> <p>Cele: j/w</p>	<p>Uczeń bada spójność grafu, rozumie sortowanie topologiczne, cykl Eulera wyznacza minimalne drzew rozpinające rozumie sposób konstrukcji struktury zbiorów rozłącznych. (Find and Union)</p> <p>omawia i rozumie skojarzenia w grafach dwudzielnych (algorytm w czasie $O(VE)$)*</p>	<p>13.1.Profesor Szu (OI13) 13.2.Plan budowy autostrad (ONTAK2011) 13.3.Bajtocja OIG1</p> <p>Zadanie domowe: 13.4.Przemytnicy (OI10) 13.5. Kolonie (szkopul.edu.pl – KI Staszic)</p>
14	<p>Podstawowe abstrakcyjne struktury danych i ich implementacje</p> <p>Cele:</p>	<p>Stos Kolejka</p>	<p>14.1.Wyrażenia nawiasowe (autor nieznany) 14.2. Stwory (autor nieznany)</p>

	<ul style="list-style-type: none"> – Poznanie podstawowych struktur abstrakcyjnych wykorzystywanych w zadaniach olimpijskich 	<p>Kopiec binarny:</p> <ul style="list-style-type: none"> – Wstawianie elementu $O(\log n)$ – Usuwanie elementu maksymalnego $O(\log n)$ – Liniowe tworzenie nowego kopca z listy elementów 	<p>14.3. K-wyważone drzewa 14.4. Taśmy (OIG5)</p> <p>Zadanie domowe: 14.5. Plakatowanie (OI16)</p>
15	<p>„Ile jest takich liczb że..” Permutacje i kombinacje. Cele:</p> <ul style="list-style-type: none"> – Podstawy kombinatoryki 	<p>podzbiory, permutacje, kombinacje (symbol Newtona), symbol Newtona na dużych liczbach, Obliczanie współczynników dwumianowych Trójkąt Pascala</p>	<p>15.1. Liczba Liczb (OIG8) 15.2. Gitara (OIG8) 15.3. Bankiet (OIG1) 15.4. Kamyczki (OIG8) 15.5. Monety (OIG5)</p> <p>Zadanie domowe: 15.6. Permutacja (OI15)</p>

8. Literatura i inne zasoby edukacyjne.

J. Tomaszewicz „Zaprzyjaźnij się z algorytmami”

P. Stańczyk „Algorytmika praktyczna nie tylko dla mistrzów”

W. Lipski „Kombinatoryka dla programistów”

Zadania archiwalne i tzw błękitne książeczki Olimpiady Informatycznej (oi.edu.pl)

T. Cormen „Wprowadzenie do algorytmów”

K. Diks, T. Idziaszek, J. Łącki, J. Radoszewski „W poszukiwaniu wyzwań”, a także „W poszukiwaniu wyzwań 2”

K. Diks, T. Idziaszek, J. Łącki, J. Radoszewski „Przygody Bajtazara. 25 lat Olimpiady Informatycznej”

algorytmika.wikidot.com

Internetowe zasoby zadań algorytmicznych (szkopuł.edu.pl, pl.spoj.com, [Codeforces](http://Codeforces.com))