Nazwa przedmiotu							
Algorytmy optymalizacji sieci Network optimization algorithms							
Dyscyplina							Oznaczenie przedmiotu
Informatyka							2S_IS1_AOS
Rodzaj przedmiotu	Stopień studiów	Tryb studiów		Język zajęć		Rok	Semestr
do wyboru	2	stacjonarne		polski		I	II
Rodzaj zajęć		Wyk.	Ćw.	Lab.	Sem.	Proj.	Liczba punktów ECTS
Liczba godzin w semestrze		15	0	30	0	0	4
Koordynator	Dr hab. inż. Andrey Grishkevich, prof. PCz. a.grischkevich@el.pcz.czest.pl						
Prowadzący	Dr hab. inż. Andrey Grishkevich, prof. PCz. a.grischkevich@el.pcz.czest.pl						

I. KARTA PRZEDMIOTU

Cel przedmiotu

C1. Wiedza na temat podstawowych pojęć i metod optymalizacji dyskretnej oraz umiejętność ich stosowania do rozwiązywania problemów optymalizacji sieci.
C2. Znajomość efektywnych metod reprezentacji maszynowej sieci i komputerowych realizacji różnych działań na krawędziach i wierzchołkach sieci.
C3. Znajomość algorytmu znajdywania najkrótszej drogi w sieci.
C4. Znajomość algorytmu znajdywania maksymalnego przepływu w sieci.
C5. Znajomość algorytmu znajdywania minimalnych przekrojów w sieci.
C6. Umiejętność komputerowej implementacji algorytmów optymalizacji sieci.

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1. Wiedza z zakresu matematyki: Kombinatoryka. Grafy. Algorytmy grafowe. Matematyka dyskretna. Optymalizacja funkcji.
2. Wiedza z zakresu programowania: Złożoność algorytmów. Język programowania C++.
3. Umiejętność obsługi komputera i pracy z INTERNET-em.
4. Znajomość języka angielskiego w stopniu wystarczającym do czytania komunikatów programów komputerowych.

Efekty kształcenia

EK1. Student zna podstawy optymalizacji dyskretnej
EK2. Student zna reprezentację maszynową sieci
EK3. Student zna algorytm: 1) znajdywania najkrótszej drogi w sieci; 2) znajdywania maksymalnego przepływu w sieci

Treści programowe: wykłady	Liczba godzin
W 1 – C++. Visual Studio. Biblioteki standartowe C++ (<list>, <stack>.). Analiza programów. – Abstrakcyjne struktury danych. Lista. Stos. – Elementy kombinatoryki. Twierdzenie o mnożeniu. Permutacje. Wariacje. Kombinacje. – Optymalizacja. Ekstremum, maksimum, minimum, ekstremum lokalne, ekstremum globalne. Postawiane pojęcia optymalizacji dyskretnej. Programowanie matematyczne. Funkcja celu. Warunki ograniczające. – Analiza sprawności algorytmów. Algorytm. Złożoność obliczeniowa algorytmów. Rzędy wielkości i porównywanie funkcji, notacje O i o. Funkcje złożoności. Problemy obliczeniowo trudne: NP-zupełność, nierozstrzygalność. Wnioski.	1
W 2 – Podstawowe pojęcia teorii grafów i sieci. Grafy i sieci. Grafy i łamigłówki. Definicje grafu. Sąsiedztwo. Typy grafów. Izomorfizm. Automorfizm. Podgrafy. Spójność. Grafy planarny. Grafy eulerowskie. Grafy hamiltonowskie. – Optymalizacja na grafach i sieciach. Zadanie chińskiego listonosza. Problem komiwojażera. Twierdzenie o czterech barwach.	2
W 3 – Algorytmy na grafach i sieciach. Reprezentacja maszynowa grafu. Algorytmy przechodzenia grafu i sieci. Metoda oznaczeń. Przeszukiwanie grafu w głąb. Przeszukiwanie grafu wszerz. Złożoność algorytmów przeszukiwania. – Algorytm szukania komponent spójności. Znajdowanie składowych dwuspójnych.	2
W 4 – Drzewa grafów. Właściwości drzew. Minimalne (maksymalne) drzewo rozpinające. Zastosowanie minimalnych (maksymalnych) drzew rozpinających. Algorytm (Kruskala, Prima) znajdywania minimalnego (maksymalnego) drzewa rozpinającego sieci. Przykład.	2
W 5 – Drogi (ścieżki) i cykle w grafie. Długość ścieżki. Zastosowanie minimalnych ścieżek. Najkrótsze ścieżki z jednym źródłem. Algorytm Dijkstry znajdywania najkrótszej drogi. Metoda oznaczeń Dijkstry. Przykład. Złożoność algorytmu Dijkstry.	2

W 6 – Przepływy w sieciach. Maksymalny przepływ. Przekroje w sieciach. Minimalny przekrój. Zastosowanie przepływów i przekrojów. Twierdzenie Forda-Fulkersona o maksymalnym przepływie i minimalnym przekroju. Metoda oznaczeń Forda-Fulkersona. Algorytm znajdywania maksymalnego przepływu. Przykład. Postępy w tworzeniu algorytmów maksymalnego przepływu. Wiele źródeł i odpływów. Przepływy o minimalnym koszcie.	2
W 7 – Minimalny przekrój. Właściwości minimalnych przekrojów. Krata dystrybutywna minimalnych przekrojów. Przeliczanie minimalnych przekrojów. Kodowanie kraty dystrybutywnej minimalnych przekrojów. Algorytm określenia zbioru minimalnnych dwuelementowych przekrojów grafu skierowanego. Przykład.	2
W 8 – Skojarzenia w grafach. Pokrycia wierzchołkowe i krawędziowe. Problem znajdywania maksymalnego skojarzenia w grafie. Skojarzenia w grafie dwudzielnym. Twierdzenie (Konig, Egervary). Algorytm znajdywania skojarzenia o maksymalnej liczności metodą Hopcrofta-Karpa. Metoda węgierska.	1
W 9 – Wnioski i podsumowania.	1
SUMA	**15**

Treści programowe: laboratorium	Liczba godzin
L 1 – Visual Studio. C++. Komunikowanie z użytkownikiem. Grafy i łamigłówki, współczesne rozwiązania komputerowe łamigłówek na podstawie istniejących programów komputerowych (użycie algorytmów teorii grafów i sieci).	3
L 2 – Grafy. Reprezentacji grafów. Przeszukiwanie grafu w głąb. Przeszukiwanie grafu w szerz (algorytm falowy).	5
L 3 – Algorytm szukania komponent spójności. Znajdowanie składowych dwuspójnych.	4
L 4 – Drzewa. Minimalne (maksymalne) drzewo rozpinające. Algorytm (Kruskala, Prima).	3
L 5 – Drogi (ścieżki). Najkrótsze ścieżki. Algorytm Dijkstry.	3
L 6 – Przepływy w sieciach. Metoda oznaczeń Forda-Fulkersona. Algorytm znajdywania maksymalnego przepływu. Test formujący.	3
L 7 – Minimalny przekróje. Algorytm określenia zbioru minimalnnych dwuelementowych przekrojów grafu skierowanego. Krata dystrybutywna minimalnych przekrojów.	3
L 8 – Skojarzenia w grafach. Skojarzenia w grafie dwudzielnym. Metoda węgierska.	3
L 9 – Odbiór obowiązkowego zestawu zadań. (Test zaliczeniowy.)	3
SUMA	**30**

Narzędzia dydaktyczne

1. Wykład z prezentacją multimedialną. Rzutnik komputerowy wraz z ekranem. Tablica klasyczna.
2. Instrukcje do ćwiczeń laboratoryjnych w postaci plików .doc, .docx, .pdf, .jpg, .txt, xls, .xlsx, .xlsm, .zip, .rar, .cpp, .exe.
3. Komputery z systemem operacyjnym Windows 7/8/10 i zainstalowanym środowiskiem programowania Microsoft Visual Studio 2015/2017, pakietem Microsoft Office 2007/2010/2013/2016 oraz przeglądarką plików .pdf, .html.
4. Podręczniki i skrypty.
5. Internet.

Sposoby oceny efektów kształcenia (F – ocena Formująca, P – ocena Podsumowująca)

F1. Aktywność na zajęciach (obecność, dyskusja, praca, wykonanie testów).
P1. Wykonanie obowiązkowego zestawu zadań w trakcie zajęć laboratoryjnych (laboratorium).
P2. Egzamin (wykłady).

Obciążenie pracą studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z prowadzącym	45
Zapoznanie się ze wskazaną literaturą	15
Przygotowanie do zajęć	10
Przygotowanie do egzaminu	15
Przygotowanie sprawozdań	15
Sumaryczna liczba godzin/punktów ECTS dla przedmiotu	**100 / 4 ECTS**

Wykaz literatury podstawowej i uzupełniającej

1. Lipski W. : Kombinatoryka dla programistów. WNT, Warszawa.
2. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C.: Wprowadzenie do algorytmów. WNT, Warszawa.
3. Sedgewick R.: Algorytmy w C++. Grafy. Oficyna Wydawnicza READ ME, Warszawa.
4. Ford L.R, Fulkerson D.R.: Przepływy w sieciach. Warszawa, Państwowe wydawnictwo naukowe.
5. Grebosz J.: Symfonia C++ standard. Programirowanie w języku C++ orientowane obiektowo. Edition, Kraków.
6. Grishkevich A.A., Hudym V.I., Kruczinin A.M., Sawicki A.: Zagadnienia energetyczne wybranych współczesnych urządzeń i systemów elektrostalowniczych // Seria Monografie. – Nr. 195. - Pod redakcją A.Sawickiego. – Częstochowa: Wydawnictwo Politechniki Częstochowskiej, 2010.

Macierz realizacji efektów kształcenia					
Efekt kształcenia	Odniesienie efektu do efektów kształcenia dla dyscypliny naukowej Informatyka*	Cele przedmiotu	Forma zajęć	Narzędzia dydaktyczne	Sposób oceny
EK1	KINF2_W01 KINF2_W09	C1, C2	W, Lab	1,2,3,4,5	F1, P1, P2
EK2	KINF2_W08 KINF2_W09	C1, C2	W, Lab	1,2,3,4,5	F1, P1, P2
EK3	KINF2_W09 KINF2_U01 KINF2_U04 KINF2_U05	C1, C2, C3, C4, C5, C6	W, Lab	1,2,3,4,5	F1, P1, P2

* – wg załącznika

II. FORMY OCENY – SZCZEGÓŁY

Ocena	Efekty
EK1	**Student zna podstawy optymalizacji dyskretnej**
2	Student nie zna podstaw optymalizacji dyskretnej.
3	Student potrafi sformułować problem w formie optymalizacji kombinatorycznej.
3.5	Student potrafi sformułować sposób rozwiązania problemu na podstawie pełnego przebierania i wskazać sposoby przyspieszenia podobnego algorytmu.
4	Student potrafi oddzielić heurystyczny i dokładny algorytmy rozwiązywania problemu optymalizacji kombinatorycznej.
4.5	Student potrafi ocenić złożoność algorytmu.
5	Student zna efektywnie sposoby komputerowej realizacji algorytmu.
EK2	**Student zna reprezentację maszynową sieci**
2	Student nie zna reprezentacji maszynową sieci.
3	Student potrafi sformułować łamigłówki (problem Eulera o przejściu 7 mostów Królewca, problem rysowania figur geometrycznych bez odrywania ołówka od papieru i bez przecinania linii (szabla Mahometa, gwiazda Dawida), trzy domy – trzy studnie, Hamilton – wokół świata, obejście koniem szachownicy, przejście labiryntu, …) jako problemy optymalizacji w grafach i sieciach.
3.5	Student zna różne sposoby reprezentacji maszynowej sieci i potrafi ocenić złożoność tej metody.
4	Student zna algorytmy przechodzenia grafu i sieci (przeszukiwanie grafu w głąb, przeszukiwanie grafu w szerz (algorytm falowy)).
4.5	Student potrafi efektywnie zaimplementować algorytmy przechodzenia grafu.
5	Student potrafi zaimplementować algorytmy szukania komponent spójności, znajdywania składowych dwuspójnych, znajdywania drzewa, drogi Eulera.
EK3	**Student zna algorytm: 1) znajdywania najkrótszej drogi w sieci; 2) znajdywania maksymalnego przepływu w sieci**
2	Student nie zna algorytmu.
3	Student potrafi sformułować wartościowe zadanie w formie zadania znajdywania: 1) najkrótszej drogi w sieci; 2) maksymalnego przepływu w sieci.
3.5	Student potrafi wyjaśniać podstawowe kroki algorytmu: 1) znajdywania najkrótszej drogi w sieci (metoda oznaczeń Dijkstry); 2) znajdywania maksymalnego przepływu w sieci (metoda oznaczeń Forda-Fulkersona).
4	Student zna sposoby implementacji kroków algorytmu w postaci programy komputerowego.
4.5	Student może ocenić złożoności różnych implementacji możliwych kroków algorytmu i wybrać najbardziej odpowiadające bieżącemu zadaniu, zna postępy w tworzeniu algorytmów (wie jak przyspieszyć działanie algorytmu).
5	Student potrafi efektywne zaimplementować algorytm w postaci programy komputerowego.

III. INNE PRZYDATNE INFORMACJE O PRZEDMIOCIE

1. Wszelkie informacje dla studentów na temat planu zajęć dostępne są na tablicy ogłoszeń oraz na stronie el.pcz.pl.
2. Informacje na temat warunków zaliczania zajęć przekazywana jest studentom podczas pierwszych zajęć.
3. Instrukcje do zajęć laboratoryjnych i treści wykładów będą umieszczane na wskazany przez prowadzącego adres poczty elektronicznej. Przejrzenie instrukcji wymaga zainstalowania oprogramowania czytającego/interpretacyjącego pliki .doc, .docx, .pdf, .jpg, .txt, xls, .xlsx, .xlsm, .zip, .rar, .cpp, .exe.
4. Zajęcia laboratoryjne będą odbywać się w sali IK2 Wydziału Elektrycznego lub równoważnej.