

# IN440 Informatica 5: Ottimizzazione Combinatoria

## A.A. 2020/2021

Prof. Marco Liverani

### Algoritmi su grafi e reti di flusso

#### 1. Teoria dei grafi

Grafo, grafo orientato, albero, albero libero e con radice, connessione, connessione forte, aciclicità; isomorfismi tra grafi, planarità, Teorema di Kuratowski, formula di Eulero; colorazione di grafi, cammini euleriani, circuiti hamiltoniani.

#### 2. Teoria degli algoritmi e dell'ottimizzazione

Richiami sugli algoritmi e sulla programmazione strutturata; complessità computazionale di un algoritmo, classi di complessità per problemi, le classi P, NP, NP-completo, NP-hard; problemi di decisione, di ricerca, di enumerazione e di ottimizzazione; problemi di programmazione non lineare, di programmazione convessa, di programmazione lineare e di programmazione lineare intera; problemi di ottimizzazione combinatoria.

Richiamo sugli elementi di calcolo combinatorio, algoritmi per la generazione dell'insieme delle parti di un insieme finito, calcolo delle permutazioni e delle combinazioni degli elementi di un insieme, calcolo del coefficiente binomiale; il problema dei "quadrati latini" e il gioco del Sudoku; un algoritmo ricorsivo per la soluzione del gioco.

#### 3. Problemi di ottimizzazione su grafi e reti di flusso

Visita di grafi, verifica di proprietà fondamentali di un grafo: connessione, connessione forte, presenza di cicli. Algoritmi per la visita di grafi, calcolo delle componenti fortemente connesse, alberi binari di ricerca, ordinamento topologico di un grafo orientato aciclico.

Il problema della costruzione di un albero ricoprente di peso minimo (*minimum spanning tree*), algoritmo di Kruskal, algoritmo di Prim, algoritmo di Sollin-Boruvka, formulazione del problema in termini di Programmazione Lineare Intera.

Ricerca di cammini minimi su un grafo pesato; cammino minimo con sorgente singola, algoritmo di Dijkstra, algoritmo di Bellman-Ford; cammino minimo tra tutte le coppie di vertici del grafo, algoritmi di programmazione dinamica, algoritmo di Floyd-Warshall, calcolo della chiusura transitiva di un grafo.

Reti di flusso e calcolo del flusso massimo su una rete, Teorema del flusso massimo e taglio minimo, algoritmo di Ford-Fulkerson, algoritmo di Edmonds-Karp, algoritmi di preflusso, algoritmi "push-relabel".

Problemi di partizionamento di grafi, alberi e cammini, problemi per il partizionamento ottimo di alberi e cammini in  $p$  componenti connesse, funzioni obiettivo, tecniche algoritmiche per la soluzione di questa classe di problemi (programmazione dinamica, programmazione lineare, *shifting*).

Problema del matrimonio stabile (*stable marriage problem*), definizione del problema e del criterio di stabilità del matching, applicazioni, algoritmo di Gale e Shapley.

Codici di Huffman, definizione del problema, presentazione dell'algoritmo per la costruzione del codice, esempi.

Algoritmi approssimati per problemi NP-completi, definizione di rapporto di approssimazione e algoritmi  $\rho(n)$ -approssimati; esempi applicati ai problemi *Vertex Cover* e *Set Cover*.

Laboratorio di programmazione per l'implementazione degli algoritmi mediante programmi in linguaggio Python e con il software Mathematica.

## TESTI CONSIGLIATI

- [1] T. H. CORMEN, C. E. LEISERSON, R. L. RIVEST, C. STEIN, *Introduzione agli algoritmi*. McGraw–Hill, (Terza edizione, 2010).  
 [2] M. LIVERANI, *Dispense del corso IN440 – Ottimizzazione Combinatoria*. (2021).

## BIBLIOGRAFIA SUPPLEMENTARE

- [3] C. H. PAPADIMITRIOU, K. STEIGLITZ, *Combinatorial Optimization. Algorithms and Complexity*. Dover Publications, (1998).  
 [4] R. J. TRUDEAU, *Introduction to Graph Theory*. Dover Publications, (1993).  
 [5] R. DIESTEL, *Graph Theory*. Springer, (2000).  
 [6] D. E. KNUTH, *Stable Marriage and Its Relation to Other Combinatorial Problems*. AMS, (1997).  
 [7] S. PEMMARAJU, S. SKIENA, *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Cambridge University Press, (2003).  
 [8] A. KELLEY, I. POHL, *C, didattica e programmazione*. Pearson – Addison Wesley, (Quarta edizione, 2004).  
 [9] M. LIVERANI, *Qual è il problema? Metodi, strategie risolutive, algoritmi*. Mimesis, (2005).  
 [10] M. LIVERANI, *Programmare in C*. Esculapio, (Seconda edizione, 2013).

## MODALITÀ D’ESAME

- valutazione in itinere (“esoneri”)		<input type="checkbox"/> SI	<input checked="" type="checkbox"/> NO
- esame finale	scritto	<input type="checkbox"/> SI	<input checked="" type="checkbox"/> NO
	orale	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO
- altre prove di valutazione del profitto (meglio descritte sotto)		<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO

L’esame orale, in cui sono discussi alcuni degli algoritmi trattati nel corso, è preceduto dalla discussione di una tesina scritta, su un argomento assegnato dal docente, in cui è richiesto, tra l’altro, di implementare in linguaggio Python gli algoritmi tratti da un articolo scientifico di ricerca o da un libro di testo, nell’ambito della risoluzione di problemi di ottimizzazione combinatoria.

Per informazioni di maggiore dettaglio sugli argomenti trattati nel corso, per ulteriori indicazioni bibliografiche e per scaricare le dispense distribuite durante le lezioni, si deve fare riferimento al sito web del corso disponibile all’indirizzo <http://www.mat.uniroma3.it/users/liverani/IN440/> o al materiale pubblicato sul canale Microsoft Teams del corso (dispense, slide, registrazioni video delle lezioni, ecc.).