

**PROGRAMMA LEZIONI ASD 2017-2018 (II SEMESTRE)**  
**PROF. CLEMENTI**

Lezione n. 1 (8-3-18): 9.00-11.00 ( $\cong$  2.5 ore acc)

1. Introduzione generale al II modulo, modalità dell'esame.
2. Def. del problema dei cammini minimi come problema di ottimizzazione.
3. Proprietà dei cammini minimi e dei loro sottocammini.
4. Algoritmo di Dijkstra per cammini minimi.
5. Correttezza e Complessità dell'Algoritmo.
6. Libro: Demetrescu, Finocchi, Italiano, *Algoritmi e Strutture Dati*, McGraw-Hill.
7. Slides del Dr. Gualà (file Dijkstra2017).

Lezione n.2 (12-3-18): ( $\cong$  2.5 ore acc)

1. IL problema del MST
2. Definizione formale come probl. di ott.
3. Applicazioni principali (cenni)
4. Proprietà del Taglio e MST
5. Proprietà del Ciclo e MST
6. Proprietà dell'intersezione tra Cicli e Tagli
7. Tre approcci Greedy
8. l'Algoritmo di Prim (Visita)
9. Correttezza dell'algoritmo di Prim mediante la Propr. 4
10. Implementazione dell'Algoritmo
11. Esempio di esecuzione
12. Analisi della complessità temporale
13. Testo: Keliberg & Tardos, Algorithm Design
14. Slides del Prof. Clementi (file mst-andy-lesson1.pdf)

Lezione n.3 (14-3-2018) ( $\cong$  2.50 ore acc.) (**Esercitazione**)

1. La relazione tra Shortest Path Tree (SPT) e Minimum Spanning Tree (MST)
2. Esempi di grafi che evidenziano la differenza
3. Algoritmo  $A(G,e)$  per la verifica dell'appartenenza di un fissato arco ad un MST di un grafo pesato connesso (Es. n 3, p. 187 di [KT]).
4. Dimostrazione della correttezza ed analisi della complessità dell'Algoritmo  $A(G,e)$

Lezione n. 4 (19-03-2018) ( $\cong$  2.5 ore acc.)

1. Riepilogo Algoritmo  $A(G,e)$
2. La Struttura Dati Union-Find
3. Definizione delle operazioni ammesse
4. Esempi
5. Soluzione mediante Foreste
6. Una soluzione non efficiente
7. La struttura Quick-Find
8. Analisi della complessità ammortizzata della struttura Quick-Find
9. Applicazione: L'Algoritmo di Kruskal K
10. Implementazione di K mediante Quick-Find
11. Analisi della complessità e della correttezza di K

Lezione n. 5 (22-03-2018) (2.5 ore acc)

1. Introduzione ai problemi di scheduling
2. Descrizione tecnica Greedy
3. Il problema Interval Scheduling
4. Approcci Greedy
5. L'Algoritmo Greedy ottimale basato sul finish time
6. Analisi della complessità
7. Dimostrazione di ottimalità: "Greedy stays ahead"
8. La riduzione del problema al problema Max Independent Set su (Interval) Graphs
9. Esercizi per casa

Lezione n. 6 (26-03-2018) (2.5 ore acc)

Il problema Interval Partitioning  
il concetto di Depth(I) di una istanza I  
Un utile lower bound per Depth(I)  
Un algoritmo Greedy ottimale  
Implementazione ed analisi della complessità  
Dimostrazione di ottimalità: confronto con il lower bound su depth(I).

Lezione n. 7 (29-03-2018) (2 ore acc)

Esercitazione sul metodo greedy e sulle sue tecniche di analisi  
L'esempio del grafo con i pesi al quadrato  
L'esercizio del minimizzare il n. di tappe di un percorso lineare  
Considerazioni generali sull'approccio greedy

Lezione n. 8 (05-04-18) (2.50 acc)

- Maximum Latency: definizione del problema di ottimizzazione
- Approcci greedy
- L'algoritmo greedy deadline-first D
- Struttura dell'ottimo
- Ottimalità dell'algoritmo D
- Riepilogo dei tools generali di progettazione ed analisi di algoritmi greedy
- Il problema del CLUSTERING
- Motivazioni e considerazioni generali.
- Una possibile definizione formale di problema in termini di spazi metrici e k-partizioni.

Lezione n. 9 (09-04-18) (2.50) acc

- Il problema del k-clustering: Definizione formale & Esempi
- Il modellamento dell'istanza come grafo completo pesato
- L'approccio Greedy basato sul MST e l'algoritmo di Kruskal
- Dimostrazione dell'ottimalità dell'Algoritmo Greedy
- Il problema della Compressione Dati
- Cenni storici, motivazioni, applicazioni moderne
- Il problema della codifica: definizione informale

- La codifica binaria di un testo a lunghezza variabile e la sua inefficienza su testi ridondanti
- I codici prefissi: definizione
- 

#### Lezione n. 10 del 12/04/18 (2.50 ore)

- Compressione dati mediante codici prefissi: definizione formale
- Rappresentazione di codici prefissi mediante alberi binari etichettati
- La funzione  $ABL(C)$  definita su codici ed alberi associati
- Progetto di algoritmi greedy per la costruzione di codici ottimali
- La struttura dell'ottimo: alberi full
- l'approccio TOP-DOWN (Codici di Shannon-Fano)

#### Lezione n. 11 del 16/04 (2.50 ore)

- Compressione dati (Riepilogo lezione precedente)
- La struttura dell'ottimo: proprietà
- L'approccio BOTTOM-UP
- Algoritmo e codici di Huffman
- Esempi di esecuzione dell'Algoritmo di Huffman
- Analisi della complessità dell'Algoritmo
- Implementazione efficiente dell'Algoritmo mediante struttura Heap

#### Lezione n. 12 del 19/04 (2.75 ore)

- La programmazione dinamica
- cenni storici ed applicazioni principali
- Descrizione informale
- Il problema Weighted Interval Scheduling
- Non ottimalità di alcuni approcci greedy
- Un possibile ordinamento dell'istanza e relativi sottoproblemi
- La struttura dell'ottimo  $OPT(j)$
- Formulazione ricorsiva di  $OPT(j)$
- Un algoritmo di programmazione dinamica ricorsivo non efficiente: analisi del caso peggiore.
- Un algoritmo iterativo efficiente: analisi del caso peggiore.

#### Lezione n. 13 del 23/04 (2.75)

- Il problema Weighted Interval Scheduling (continua)
- Come calcolare la soluzione ottima da  $OPT(j)$
- La versione ricorsiva efficiente dell'algoritmo di programmazione dinamica: analisi della complessità mediante la misura progresso  $\Phi$ .
- Il problema Segmented Least Squares
- Introduzione informale e definizione rigorosa.
- Esempi
- Formulazione del problema in termini di programmazione dinamica

- L'algoritmo ottimale: versione iterativa ed analisi della complessità
- 

Lezione n. 14 del 26/04 (2.50 ore)

- Programmazione dinamica: ripasso degli esempi precedenti e valutazioni generali sulle condizioni di applicabilità
- Programmazione dinamica: Il problema della Bisaccia.
- Un approccio di programmazione dinamica con un parametro che non funziona
- Un approccio a due parametri che funziona
- Analisi dell'ottimalità
- Implementazione dell'algoritmo ricorsivo
- Analisi della complessità
- Gli algoritmi pseudopolinomiali

Lezione n 15 del 03/05: Esercitazioni del Dr. Gualà

- programmazione dinamica: calcolare cammini minimi in grafi con pesi negativi
- cammini minimi in grafi pesati
- problema dei cammini minimi a singola sorgente (o singola destinazione)
- grafi con pesi negativi: false partenze
- un algoritmo di programmazione dinamica: algoritmo di Bellman-Ford (BF) con complessità temporale  $O(mn)$  e spaziale  $O(n^2)$
- ridurre la memoria a  $O(n)$
- modificare l'algoritmo di BF per rilevare cicli di peso negativo

Lezione n. 16 del 07/05: Esercitazioni del Dr. Gualà

Esercitazione sulla programmazione dinamica.

I due esercizi discussi possono essere trovati nelle seguenti note:

[http://www.mat.uniroma2.it/~guala/discussi\\_2017.pdf](http://www.mat.uniroma2.it/~guala/discussi_2017.pdf)

Lezione n. 17 del 10/05 (3.15 ore)

- Programmazione dinamica: il problema del Sequence Alignment
- Modellazione del Problema e principali applicazioni
- Un primo approccio sbagliato di Programmazione Dinamica ad una variabile
- L'approccio a due variabili corretto
- Complessità e correttezza: lo spazio utilizzato
- La trasformazione a problemi di cammini minimi
- Introduzione alla classe NP
- Problemi efficientemente verificabili: esempi
- Ripasso della Pseudopolinomialità

Lezione n. 18 del 14/05 (3.00 ore)

- Definizione formale di Certificatori efficienti
- Definizione formale della classe NP

- Relazione di NP con P ed EXP
- Esempi di problemi e di certificatori per essi
- Riducibilità polinomiale: versioni di Cook e Karp
- Proprietà della riducibilità di Karp: chiusura rispetto a P

Lezione n. 19 del 17/05 (Esercitazione del Dr. Gualà)

- Introduzione agli algoritmi randomizzati.
- Il problema del calcolo del global mini-cut.
- Algoritmo di Karger. Dimostrazione che il numero di global min-cut di un grafo di  $n$  nodi è al più  $n(n-1)/2$ .
- Testo: KT, pag 714-719.

Lezione n. 20 del 21/05 (Lezione del Dr. Gualà)

- Il problema dell'accesso ad una risorsa condivisa (Contention Resolution). Un protocollo distribuito. Analisi del protocollo.
- Un (sorprendente) algoritmo di approssimazione per il problema di max 3SAT. Analisi dell'algoritmo. Stima del numero atteso di clausole soddisfatte. Accenno al metodo probabilistico. Stima del numero atteso di tentativi prima di trovare una soluzione che soddisfa i  $7/8$  delle clausole.

Testo: KT, pag 708-714, 719-721, 724-727.

Lezione n. 21 del 24/05 (3.00 ore)

- Problemi NP-completi
- Il problema 3-SAT è in NP
- NP-completezza di 3-SAT (senza dimostrazione)
- Il problema Independent Set è in NP
- La riduzione polinomiale 3-SAT  $<$  IS: la tecnica dei GADGETS
- Il problema Vertex Cover
- Relazione tra VC e IS: equivalenza.

Lezione n. 22 del 28/05 (3.00 ore)

- Concetto di equivalenza fra problemi NP-completi (ripasso)
- Problemi di Ottimizzazione: definizione formale ed esempi
- Relazione tra problemi di ottimizzazione e loro versione decisionali
- Problemi di ottimizzazione NP-Hard
- Concetto di algoritmo approssimante: definizione formale
- Un algoritmo 2-approssimante per VC

Lezione n. 23 del 31/05/2018 (Esercitazione Dr. Scornavacca)

- Algoritmi di approssimazione
- Problema del Load Balancing
- Algoritmo 2-approssimato
- Algoritmo 3/2-approssimato
- Problema del Knapsack
- Algoritmo (1-epsilon)-approssimato.

Lezione n. 24 del 04-06-2018

- Esercitazione su programmazione dinamica
- Es 3 dell'appello del 12/02/18  
([http://www.mat.uniroma2.it/~guala/ASD\\_compito12\\_02\\_18\\_mod1.pdf](http://www.mat.uniroma2.it/~guala/ASD_compito12_02_18_mod1.pdf))
- Es 3 dell'appello del 19/09/17  
([http://www.mat.uniroma2.it/~guala/ASD\\_compito19\\_09\\_17\\_mod1.pdf](http://www.mat.uniroma2.it/~guala/ASD_compito19_09_17_mod1.pdf)).