ANNO ACCADEMICO 2018-2019
ASD (II MODULO)
PROF. ANDREA CLEMENTI
LEZIONI

1) LEZIONE 04-03-2019

1. Introduzione generale al II modulo . Conoscenze necessarie per gli studenti.
2. Linguaggio logico e matematico, pseudocodice, e nozioni fondamentali utilizzati nelle lezioni e richiesti come obiettivi formativi essenziali del corso.
3. Un argomento già visto nel I modulo, rivisitato sotto la luce dei suddetti obiettivi: L'algoritmo di Dijkstra per i cammini minimi.
4. Definizione formale del problema
5. Lo Shortest Path Tree (SPT)
6. Il Principio di Sub-Ottimalità dei cammini minimi
7. Lo schema dell'algoritmo di D.
8. Correttezza dello schema
9. Dettagli implementativi e complessità

2) LEZIONE 7-3-19

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. Testo: Keliberg & Tardos, Algorithm Design

3) LEZIONE 11-3-19

1. Correttezza dell'algoritmo di Prim mediante la Propr. 4
2. Implementazione dell'Algoritmo
3. Esempio di esecuzione
4. Analisi della complessità temporale
5. Algoritmo di Kruskal: Correttezza
6. Algoritmo di Kruskal: Complessità e gestione delle componenti connesse
7. La stuttura dati UNION & FIND (introduzione)
8. Il suo utilizzo nell'algoritmo di Kruskal

4) LEZIONE 14-3-19

1. Strutture dati per operazioni: Union&Find.
2. Implementazione mediante foreste
3. Analisi ammortizzata del tempo su una sequenza arbitraria di operazioni
4. La relazione tra Shortest Path Tree (SPT) e Minimum Spanning Tree (MST)
5. Esempi di grafi che evidenziano la differenza

5) LEZIONE 18-3-19

1. 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]).
2. Dimostrazione della correttezza ed analisi della complessità dell'Algoritmo A(G,e)
3. Il problema del Clustering: definizione formale del Pb. k-clustering
4. Soluzione efficiente del Clustering mediante MST
5. Dimostrazione di ottimalità della soluzione MST

6) 21/03/2019

1. k-clustering: Ripasso della Dimostrazione di ottimalità della soluzione MST.
2. l'approccio greedy: considerazioni generali.
3. Problemi di Scheduling.
4. Interval Scheduling: definizione del problema
5. Approcci greedy non buoni
6. Approccio greedy ottimali: finish time
7. Dimostrazione di ottimalità: greedy stays ahead

7) 25/03/2019

1. Interval Scheduling :Rivisitazione della prova di ottimalità del greedy
2. Interval Partitioning: definizione del problema
3. Interval Partitioning: analisi di approcci greedy ed esaustivi
4. Interval Partitioning: un lower bound per l'ottimo
5. Interval Partitioning: un approccio greedy ottimale
6. Interval Partitioning: dimostrazione di ottimalità del greedy
7. Esercizio: Minimizzare il numero di stazioni (the car traveling problem)

8) 04-04-2019

1. Interval Partitioning: rivisitazione della dimostrazione di ottimalità del greedy
2. Minimizing Lateness: definizione del problema
3. Approcci greedy non buoni
4. Approccio greedy ottimali: deadline time
5. Dimostrazione di ottimalità: soluzioni canoniche (no idle time)
6. Dimostrazione di ottimalità: rimozione di inversioni
7. Dimostrazione di ottimalità: exchange argument
8. Considerazioni generali sull'approccio greedy
9. Il problema della compressione dati: Introduzione

9) 08-04-2019

1. Il problema della compressione dati: Definizione formale
2. I codici prefissi, la misura ABL
3. la rappresentazione mediante alberi etichettati
4. Proprietà delle strutture ottime
5. Un primo approccio Top-Down (Shannon-Fano)
6. L'approccio Bottom-Up (Huffman)

10) 11-04-2019

1. Compressione Dati e Codici prefissi: Ripasso
2. La struttura della soluzione ottima: proprietà greedy
3. L'approccio Bottom-Up (Huffman)
4. Implementazione dell'Algoritmo di Huffman HUF in forma ricorsiva
5. Esecuzione di HUF su un esempio concreto ed osservazioni sul processo di costruzione dell'albero
6. Analisi della complessità di HUF
7. Ottimalità della soluzione generata da HUF
8. Introduzione alla Programmazione Dinamica
9. Il problema Weighted Interval Scheduling: considerazioni iniziali

11) 15-04-2019

1. Il problema Weighted Interval Scheduling (WIS): Definizione formale
2. L'approccio di programmazione dinamica per WIS
3. Ordinamento dell'istanza
4. La definizione rigorosa dei sottoproblemi WIS(j)
5. Un'equazione ricorsiva per il costo OPT(j) per WIS(j)
6. Dimostrazione dell'equazione
7. Algoritmo ricorsivo: il suo worst-case esponenziale
8. Come rimuovere la Ridondanza dei calcoli: la Memoization
9. Calcolo della Matrice/Vettore M(j)
10. Analisi della complessità della versione iterativa
11. Come costruire anche la soluzione ottima, non solo il suo costo
12. Il problema del Segmented Least Squares (SLS): introduzione e definizione formale

12) 18-04-19

1. Il problema del Segmented Least Squares (SLS): ripasso della formalizzazione
2. SLS: struttura della soluzione ottima
3. SLS: definizione corretta dei sottoproblemi per un approccio di programmazione dinamica
4. SLS: formula ricorsiva ed iterativa della soluzione ottimale
5. SLS: descrizione della matrice e dell'algoritmo
6. SLS: complessità dell'algoritmo
7. Il problema KnapSack (KS): introduzione e definizione formale
8. Un approccio greedy non ottimale
9. Un approccio errato di programmazione dinamica con un solo parametro
10. Strutturazione dell'ottimo usando sottoproblemi con due parametri
11. Verifica della struttura ricorsiva a due parametri: prova della correttezza
12. Organizzazione efficiente dei calcoli: Calcolo della Matrice 2-dimensionale
13. Pseudopolinomialità

13) 29-04-19

Esercitazioni di Riccardo Denni su Compressione Dati e Programmazione Dinamica:

1. Esecuzione dell'Algoritmo di Huffman
2. Non-ottimalità dell'algoritmo di Shannon-Fano (controesempi)
3. Altri esempi di programmazione dinamica

14) 02-05-19

1. Programmazione Dinamica: Sequence Alignment
2. Struttura di un ottimo alignment: proprietà fondamentali
3. Organizzazione efficiente dei calcoli: Calcolo della Matrice 2-dimensionale
4. Introduzione alla classe NP

15) 06-05-2019

1. NP: certificazione efficiente delle soluzioni di un problema decisione
2. NP: certificatori efficienti
3. NP: esempi di problemi decisionali che ammettono certificatori efficienti (3-SAT)
4. NP: definizione formale e confronto con le Macchine di Turing Non-Deterministiche
5. NP: asimmetria nel criterio di accettazione
6. P < NP < EXP (dimostrazione delle relazioni tra le tre classi)
7. Riduzioni polinomiali (Introduzione)

16) 09-05-19

1. Definizione di Cook-Riduzione Polinomiale (mediante Oracolo)
2. Definizione di Cook-Riduzione Polinomiale (trasformazione polinomiale)
3. Proprietà transitiva delle riduzioni polinomiali e sua ''efficienza''
4. Chiusura della classe P rispetto alle riduzioni polinomiali
5. Problemi NP-completi e loro collocazione nel caso P # NP
6. Un primo problema NP-Completo: CIRUIT-SAT
7. La riduzione CIRCUIT-SAT < 3-SAT

17) 13-05-19

1. Problemi non in NP (forse): la classe co-NP
2. La struttura P $\subseteq$ NP $\cap$ co-NP
3. Il Thm di Pratt e la dimostrazione che PRIME e' in NP $\cap$ co-NP
4. Riduzioni Polinomiali mediante tecnica dei Gadget: 3-SAT < INDEPENDENT SET

18) 16-05-19

1. Dimostrazione formale di 3-SAT < INDEPENDENT SET
2. Il problema VERTEX COVER
3. VERTEX COVER E INDEPENDENT SETS: Proprietà combinatoriche
4. Equivalenza VERTEX COVER = INDEPENDENT SET
5. Le tecnica di riduzione: *Generalization.*
6. La riduzione VERTEX COVER < SET COVER
7. Cenni sul concetto di approssimazione di problemi NP-hard

19) 20-05-2019 (Esercitazione - Riccardo Denni)

1. Ripasso concetti su NP-completezza e Riduzioni Polinomiali
2. Riduzione 3-SAT < 3-COL
3. Riduzione 3-SAT < SUBSET SUM
4. Knapsack

20) 23-05-2019

1. Gli algoritmi di approssimazione per problemi NP-Hard
2. Definizione formale di algoritmo di approssimazione
3. l'algoritmo 2-apx per Min-VC basato su Maximal Matching: progetto ed analisi
4. l'algoritmo 2-apx per Load Balancing basato sull'approccio Greedy
5. l'algoritmo (1.5)-apx per Load Balancing basato sull'approccio Greedy
6. Risultati di Apx-Hardness: introduzione
7. I' apx-hardness per Min-TSP: La Gap Technique (descrizione ed analisi)
8. Conclusione del Corso

21) 30-05-2019 (Esercitazione - Riccardo Denni)

1. Esercitazione su algoritmi di approssimazione e riduzioni:
2. Algoritmo 2 approssimante per Euclidean TSP,
3. Algoritmo 2 approssimante per Bin Packing;
4. Riduzione da Subset Sum a Knapsack.