

Diario delle lezioni di Algoritmi e Strutture Dati, modulo II, a.a. 2016/17.

1. (6/3/17). Teoria dei grafi, problemi su grafi, algoritmi su grafi. Nozioni preliminari. Grafi Euleriani. Il problema della colorazione di un grafo. Un algoritmo greedy per colorare un grafo.
2. (7/3/17). Strutture dati per rappresentare un grafo. Visite di un grafo. Visita in ampiezza (BFS): cammini minimi da una sorgente. Visita in profondità (DFS): uscire da un labirinto.
3. (13/3/17) Usi meno comuni della visita DFS. Catalogare per tipo gli archi del grafo. Individuare un ciclo in grafi diretti. Grafi diretti aciclici (DAG) e ordinamento topologico. Usare la visita DFS per trovare un ordinamento topologico di un DAG. Componenti fortemente connesse: un algoritmo lineare per calcolarle.
4. (14/03/16) Esercitazione. Primo esercizio (Probl. 1): dato un insieme di esami e un insieme di propedeuticità, progettare un algoritmo che determina quali esami fare in ogni sessione in modo da minimizzare il numero di sessioni. Secondo esercizio (Probl. 2): due robot telecomandabili si possono muovere su un grafo; in ogni istante si può spostare lungo un arco solo uno dei due robot e i robot non possono essere troppo vicini fra loro (per motivi di interferenza delle antenne); trovare la sequenza più corta di mosse che spostano i due robot dalle posizioni iniziali a due date posizioni finali. Terzo esercizio (Probl. 3): si vogliono posizionare più monete possibile su un grafo; la mossa che si può fare è mettere una moneta su un nodo libero e (necessariamente) farla scorrere lungo un arco verso un altro nodo libero; trovare un algoritmo che determina la strategia ottima.
5. (20/03/17) Cammini minimi in grafi pesati: episodio I. Il problema del calcolo dei cammini minimi a singola sorgente. Un algoritmo veloce quando il grafo ha pesi non negativi: l'algoritmo di Dijkstra.
6. (21/03/17) Cammini minimi in grafi pesati: episodio II. Ancora sul problema del calcolo dei cammini minimi a singola sorgente. Un algoritmo per grafi con pesi negativi (ma non cicli negativi): algoritmo di Bellman e Ford. Usare l'algoritmo di Bellman e Ford per rilevare un ciclo di peso negativo. Trovare i cammini minimi a singola sorgente in un DAG in tempo lineare.
7. (27/03/17). Cammini minimi in grafi pesati: episodio III. Calcolare le distanze fra tutte le coppie di nodi. Algoritmo di Floyd e Warshall e algoritmo di Johnson.
8. (28/03/17). Esercitazione. Esercizio uno (Probl. 4): dato grafo G non orientato con pesi positivi sugli archi, e dato un sottoinsieme di k nodi detti *centri*, partizionare i nodi di G in k insiemi in modo che l'insieme i contenga i nodi che sono più vicini all' i -esimo centro che ad ogni altro. Esercizio due (Probl. 5): dato un grafo orientato G con pesi positivi sugli archi e che ha un sottoinsieme di

archi detti blu, trovare il cammino di costo minimo da un certo nodo s a un certo nodo t che usa al più k archi blu.

9. (03/04/17). Mantenere efficientemente degli insiemi disgiunti: il problema Union-Find. Due approcci: QuickFind e Quick Union. Euristiche per l'operazione di Union. Stato dell'arte sul problema, una complessità che dipende dalla funzione inversa della funzione di Ackermann.
10. (04/04/17). Il problema del minimo albero di copertura (*minimum spanning tree*). Definizione del problema, motivazioni, proprietà fondamentali su cicli e tagli. L'algoritmo di Kruskal.
11. (10/04/17). Algoritmo di Prim. Il problema del calcolo dei minimi antenati comuni in un albero. Algoritmo di Tarjan: un uso interessante delle strutture dati Union-Find.
12. (11/04/17). Esercitazione. Esercizio uno (Problema 6): dato un grafo $G=(V,E,w)$ e un arco e , rispondere in tempo lineare alla domanda: l'MST di G contiene o meno l'arco e ? Esercizio due (Problema 7): trovare il miglior albero ricoprente vincolato ad avere certi nodi come nodi foglie. Esercizio tre (ordinamenti topologici e programmazione dinamica) (Probl.8): dato un dag G e due nodi s e t , calcolare in tempo lineare il numero di cammini distinti in G da s a t .
13. (20/04/2017). Dimostrare l'ottimalità degli algoritmi Greedy: la tecnica Greedy Stays Ahead [KT05, pp 116 - 124]. Applicazione della tecnica all'analisi di un algoritmo per il problema Interval Scheduling. Un problema correlato: Scheduling all Intervals. The survivors Run (Solved Exercise 1).
14. (27/04/2017). Dimostrare l'ottimalità degli algoritmi Greedy: l'Exchange Argument [KT05, pp 125-136]. Progettazione ed analisi dell'algoritmo per il problema di Minimize Lateness. Progettazione ed analisi dell'algoritmo per il problema di Optimal Caching.
15. (02/05/2017). Analisi dell'algoritmo per il problema di Optimal Caching. Compressione Dati [KT05, pp 161-176]. Codifiche a lunghezza variabile. Codici prefissi e alberi binari. La codifica di Huffman. Ottimalità della codifica di Huffman.
16. (04/05/2017). Esercitazione sugli algoritmi greedy (Esercitazione 4).
17. (08/05/2017). Esercitazione sugli algoritmi greedy (Esercitazione 5).
18. (09/05/2017). Introduzione al problema del Massimo Flusso. Algoritmo di Ford-Fulkerson. Prima analisi sulla terminazione [KT 337-345]. Massimo flusso e minimo taglio: un teorema importante. Ottimalità dell'algoritmo di Ford-Fulkerson [KT 346 - 351].

19. (15/05/2017). Richiami sull'ottimalità dell'algoritmo di Ford Fulkerson. Rendere polinomiale l'algoritmo di Ford-Fulkerson: Scaling Max-Flow Algorithm [KT 352 - 356] e il Cammino aumentante di lunghezza minima [Dispense le prof. Gambosi http://www.mat.uniroma2.it/~guala/asd_2013.pdf pag 22 -25].
20. (16/05/2017). Risolvere problemi trasformandoli in problemi di flusso: matching su grafi bipartiti, flusso massimo con sorgenti e destinazioni multiple, flusso con vincoli di capacità sui nodi.
21. (22/05/2017). Esercitazione sui problemi di flusso (Esercitazione 6).
22. (23/05/2017). 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 $n(n-1)/2$. [KT: pag 714-719].
23. (29/05/2017). Il problema dell'accesso ad una risorsa condivisa (Contention Resolution). Un protocollo distribuito. Analisi del protocollo. [KT 708 - 714]. 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. [KT 719-721, 724-727].
24. (30/05/2017). Tabelle Hash: un'alternativa ad un BST per l'implementazione di un dizionario. Scegliere un buona funzione hash. Definizione di famiglia universale di funzioni hash. Usare una famiglia universale di funzioni hash per progettare un dizionario con complessità attesa costante per operazione. Un esempio di classe universale di funzioni hash. [KT 734 - 741].