

Problem Set 2

Algoritmi e Strutture Dati – a.a. 2025/2026

Università di Roma “Tor Vergata”

Prof. Luciano Gualà

Importanti avvertenze preliminari:

1. Il gruppo deve essere formato da almeno tre persone!
2. Per lo svolgimento si consiglia fortemente l'utilizzo del template L^AT_EX disponibile nella pagina web del corso.
3. Leggere il documento *Problem Set: Istruzioni e consigli* sulla pagina web del corso.

Linee guida per la consegna. Gli elaborati dovranno essere consegnati entro domenica 25/01/2026 alle ore 23:59. Si prega di seguire le seguenti indicazioni:

1. le soluzioni dovranno essere consegnate via email a entrambi gli indirizzi `guala@mat.uniroma2.it` e `alessandrostr95@gmail.com`;
2. l'oggetto dell'email dovrà essere “Consegna problem set 2 - 2025/2026”;
3. il file dovrà essere in formato pdf;
4. il nome del file dovrà essere “PS_2_XXX.pdf” dove XXX è il primo cognome (in lower case) in ordine alfabetico dei membri del gruppo;
5. l'email dovrà essere inviata tramite l'indirizzo dello studente XXX presente nel nome del file;
6. inserire nel file nome, cognome, indirizzo email e matricola di tutti i membri del gruppo (meglio ancora se tutti i membri del gruppo sono in indirizzo nell'email della consegna).

Buon lavoro 🍌👓!

Problema 1 (*Straziota va in Oriente*)

Il dottor maestro Straziota sta per partire per l'estremo oriente, dove visiterà per qualche mese un'università straniera. Ha preparato una valigia enorme (dove ha messo anche una bottiglia di olio d'oliva, perché lui è pugliese e non sa stare senza) che vuole portare con sé. Si sa che in Italia gli studenti di dottorato non guadagnano molto, quindi Straziota nel comprare i biglietti aerei che porteranno lui e la sua enorme valigia a destinazione vuole spendere il meno possibile.

Le tratte aeree disponibili possono essere modellate attraverso un grafo diretto $G = (V, E)$ dove i nodi rappresentano gli aeroporti e un generico arco (diretto) (u, v) rappresenta un volo diretto (o *tratta*) dall'aeroporto u all'aeroporto v . Tale tratta può essere utilizzata per spostare Straziota e/o la sua valigia da u a v , ad un costo rispettivamente $w_A(u, v)$ e $w_B(u, v)$. Entrambi, Straziota e la sua valigia, partono dall'aeroporto s e devono arrivare all'aeroporto t , potenzialmente seguendo cammini diversi nel grafo G .¹

Al momento della prenotazione dei biglietti, Straziota scopre di avere due coupon per la valigia. Un coupon dà diritto ad uno sconto del 50% sul costo per spedire la valigia lungo una tratta, ma può essere utilizzato solo se si acquista *anche* il biglietto per il passeggero relativo alla stessa tratta.

Progettate un algoritmo veloce che aiuti Straziota a prenotare i biglietti dei due cammini in modo da minimizzare il costo totale.

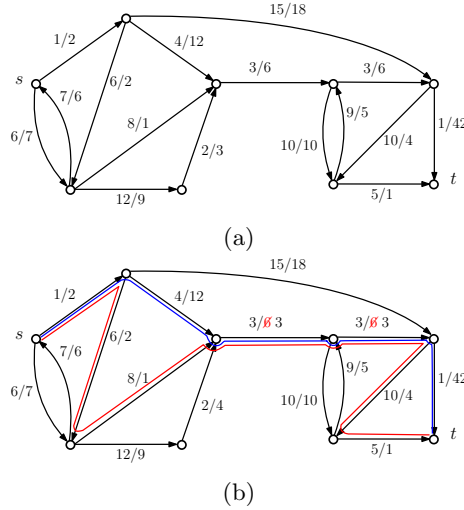


Figure 1: La figura (a) mostra il grafo $G = (V, E)$ in input, coi relativi pesi w_A (primo valore) e w_B (secondo valore). La figura (b) mostra in blu le tratte che prende il dottor maestro, mentre in rosso quelle che prende il suo bagaglio. Possiamo vedere che i coupon vengono usati in due tratte consecutive, risparmiando ben 6\$ in totale sul costo del bagaglio.

¹Entrambi i cammini, quello che seguirà Straziota e quello che seguirà la sua valigia, devono ovviamente partire da s e arrivare a t .

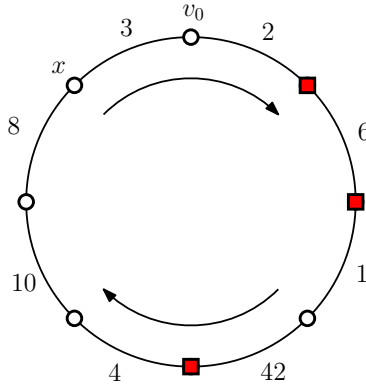


Figure 2: Esempio di istanza del problema “*Girando per il GRA*”. I quadrati rossi rappresentano le stazioni di servizio. L’output della query `proxStaz(x)` è v_1 (a distanza 5 da x), mentre l’output della query `ultimaStaz(x, 20)` è v_2 .

Problema 2 (*Girando per il GRA*)

La vostra città è circondata da un anello autostradale a scorrimento veloce. L’anello autostradale può essere percorso solo in una direzione (in senso orario) ed è diviso in n tratte, di lunghezza non omogenea. Potete modellare l’anello autostradale come un grafo G di n nodi – diciamo $v_0, v_1, \dots, v_{n-1}$ – ed n archi, dove l’arco $(v_i, v_{(i+1) \bmod n})$ ha una lunghezza w_i . Alcuni nodi del grafo ospitano delle stazioni di servizio dove è possibile fare rifornimento. Si veda la Figura 2 per un esempio.

Progettate una struttura dati di dimensione $O(n)$ che è in grado di rispondere alle seguenti query:

- `proxStaz(x)`: dato un nodo x restituisce il prossimo nodo (in senso di marcia orario) che ospita una stazione di servizio, fornendo anche la distanza di tale nodo dal nodo x . La complessità per rispondere a questa query deve essere $O(1)$.
- `ultimaStaz(x, L)`: prende in input un nodo x una distanza massima percorribile L e chiede di restituire l’ultimo nodo che ospita una stazione di servizio fra i nodi a distanza al più L da x .

Questa query serve per capire dove si trova l’ultima stazione di servizio utile dove fermarsi se si ha benzina sufficiente per percorrere ancora distanza L . Il tempo per rispondere a questa query deve essere $O(\log n)$.

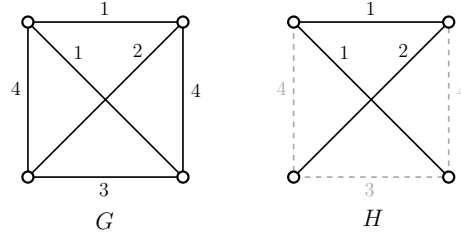


Figure 3: Sulla destra il grafo G in input, con *distanza totale* $R(G) = 24$. Sulla destra un albero H con *distanza totale* $R(H) = 26$.

Problema 3 (*Sottoreti sparse di distanza totale piccola*)

Una rete di comunicazione è modellata come un grafo non orientato e pesato $G = (V, E, w)$ di n vertici in cui ad ogni arco $e \in E$ è associato un peso non negativo $w(e)$.

Si vuole selezionare una sottorete H di G (attivando soltanto un sottoinsieme degli archi di E) da usare per far comunicare in modo efficace tutte le coppie di nodi in V . In particolare si vuole selezionare H in modo tale che la *distanza totale* in H sia piccola, dove la distanza totale di H è definita come la somma delle distanze in H di tutte le coppie *ordinate*² di vertici, ovvero

$$R(H) = \sum_{u,v \in V} d_H(u,v).$$

Ora, se non ci fossero vincoli, la cosa migliore sarebbe scegliere $H = G$. Purtroppo però, per questioni di costi di attivazione, dovete garantire che H abbia solo $n-1$ archi, ovvero sia un albero.³

Progettate un algoritmo efficiente (con complessità polinomiale in n) che calcola un albero H tale che $R(H)$ non sia troppo più grande di $R(G)$, ovvero tale che $R(H) \leq 2R(G)$.

Domanda bonus: si mostri che non è possibile garantire una qualità migliore, ovvero non è possibile garantire in generale che $R(H) \leq \alpha R(G)$ con $\alpha < 2$.

²Con *ordinate* si intende che, dati due nodi u e v , nella sommatoria è presente sia il termine $d_H(u,v)$ che il termine $d_H(v,u)$.

³In generale, si noti che per ogni sottografo H vale che $R(H) \geq R(G)$.