INFORMATICA GENERALE - I-Z

Claudia Malvenuto - Daniele A. Gewurz

Scheda esercizi n. 8

1. Siano date n attività (seminari, ecc.) e per ogni attività i ($1 \leq i \leq n$), l'intervallo temporale $[s_i, f_i)$ in cui l'attività dovrebbe svolgersi (dove s_i è il momento in cui inizia e f_i quello in cui finisce). Scrivere un algoritmo greedy che selezioni il maggior numero possibile di attività che possono essere svolte senza sovrapposizioni in un'unica aula.

2. Supponiamo di avere n file di lunghezze $l_1, \dots, l_n$ (interi positivi) che bisogna memorizzare su un disco di capacità data D. Si assuma che la somma delle lunghezze di questi file ecceda la capacità del disco. Vogliamo selezionare un sottoinsieme degli n file che abbia cardinalità massima e che possa essere memorizzato sul disco. Descrivere un algoritmo greedy che risolve il problema, dimostrarne la correttezza e valutarne la complessità.

3. Dobbiamo percorrere il tragitto che va da una località A a una località B con un'auto che ha un'autonomia di k chilometri. Lungo il percorso, a partire da A, sono presenti n distributori di benzina ciascuno distante dal precedente meno di k chilometri; l'ultimo dista meno di k chilometri da B. Sia d_i la distanza che separa il distributore i dal distributore $i+1$ per $i = 1, 2, \dots, n-1$ e sia d_n la distanza da B dell'ultimo distributore. Descrivere un algoritmo greedy che seleziona un numero minimo di distributori in cui fare rifornimento durante il viaggio, dimostrarne la correttezza e valutarne la complessità.

4. In un grafo orientato G un sottoinsieme A degli archi è detto *univoco* se non contiene 2 o più archi uscenti dallo stesso nodo. Descrivere un algoritmo greedy che preso in input un grafo G orientato con pesi positivi sugli archi, trovi un insieme univoco A di archi di G di peso massimo. Dimostrare la correttezza dell'algoritmo proposto e valutare la complessità di una sua efficiente implementazione.

5. Consideriamo il problema di dare un resto di n euro con il minor numero possibile di monete.

   (a) Si descriva un algoritmo greedy che dia un resto di n euro usando monete del valore di 0,10, 0,50, 1 e 2 euro. Dimostrare la correttezza dell'algoritmo proposto e valutare la complessità di una sua efficiente implementazione.

   (b) Si generalizzi l'algoritmo del punto precedente al caso in cui i valori di monete disponibili siano $c_1, c_2, \dots, c_k$ tali che c_i divide c_{i+1} per $1 \leq i < k$. Provare la correttezza dell'algoritmo proposto e valutarne la complessità.

   (c) Si mostri un insieme di valori di monete per cui l'algoritmo proposto al punto precedente non è corretto.

6. Si considerino n oggetti ed uno zaino. L'oggetto i ha peso w_i e valore v_i mentre la capacità dello zaino è C. Se nello zaino viene inserita una frazione x_i dell'oggetto i, ove $0 \leq x_i \leq 1$, allora si ottiene un profitto $v_i \cdot x_i$ e la capacità dello zaino diminuisce di $w_i \cdot x_i$. Si desidera riempire lo zaino massimizzando il profitto totale.

   (a) Descrivere un algoritmo greedy che risolva il problema.

   (b) Provare la correttezza dell'algoritmo proposto e valutarne la complessità (la complessità non dovrebbe superare $O(n \log n)$).