



Esercitazione di laboratorio n. 10

(Le soluzioni verranno valutate in Laboratorio. Solo se caricate nella Sezione Elaborati del Portale entro e non oltre il 16/12/2014 concorreranno all'assegnazione dei punti supplementari)

Esercizio n. 1: Problemi di ottimizzazione

Dato un insieme S , una collezione $\mathcal{S} = \{S_i\}$ di sottoinsiemi non vuoti forma una **partizione** di S se e solo se valgono entrambe le seguenti condizioni:

- i sottoinsiemi sono a coppie disgiunti, cioè $\forall S_i, S_j \in \mathcal{S}$ con $i \neq j$ $S_i \cap S_j = \emptyset$ e
- la loro unione è S , cioè $S = \bigcup_{S_i \in \mathcal{S}} S_i$.

Si supponga che l'insieme S contenga n elementi (cioè $\text{card}(S) = n$) e che lo si voglia partizionare in k sottoinsiemi (cioè $\text{card}(\mathcal{S}) = k$) in modo ottimo per risolvere il seguente problema:

- sia dato un elenco di risorse disponibili letto da un primo file il cui nome è passato sulla riga di comando. Ogni risorsa sia identificata in maniera univoca da una stringa di al più 20 caratteri. Nel file non è riportato il numero delle risorse k
- sia dato un elenco di compiti da svolgere letto da un secondo file il cui nome è passato sulla riga di comando. Ogni compito sia identificato in maniera univoca da una stringa di al più 10 caratteri. A ogni compito è associato un intero che ne identifica la difficoltà. Nel file non è riportato il numero di compiti da svolgere n .

Si scrivano due programmi C che, letti i 2 file, determinino la soluzione ottima del problema di assegnazione dei compiti alle risorse con i seguenti vincoli:

- ogni compito viene svolto da una sola risorsa
- nessuna risorsa deve restare inutilizzata
- criterio di ottimalità: nella partizione è minima la differenza tra il "carico di lavoro" della risorsa più utilizzata e quello della risorsa meno utilizzata. Sia definito il carico di lavoro assegnato a ogni risorsa come $\sum_{k=0}^z a_k$, dove z è il numero di compiti assegnati a quella risorsa e a_k la difficoltà di ognuno di essi.

Il primo programma sia basato su un'enumerazione esaustiva dello spazio delle possibilità e ritorni quindi la soluzione ottima, il secondo segua un approccio greedy e quindi non ritorni necessariamente una soluzione ottima. Sia a carico dello studente la definizione della funzione di costo/appetibilità nell'approccio greedy.

Esempio: siano il file dei compiti e il file delle risorse i seguenti:

compiti.txt	risorse.txt
job_1 6	risorsa_A
job_2 12	risorsa_B
job_3 7	
job_4 8	
job_5 6	

Una possibile soluzione è il seguente partizionamento degli incarichi:

risorsa_A {job_1, job_3, job_5}, carico complessivo: 19

risorsa_B {job_2, job_4}, carico complessivo: 20