

05.05.2017

Esame di Fondamenti Informatica per Ingegneria Gestionale - A.A. 2016/2017

Durata 1h45' - Compito A

Istruzioni (leggere attentamente)

Nota importante: la mancata osservanza delle seguenti regole può comportare la perdita di informazioni necessarie alla valutazione dell'esame.

Cartella di esame e registrazione dei dati dello studente

La cartella Esame contenente il compito da svolgere si trova sul Desktop. Entrare nella cartella e, prima di iniziare il compito, eseguire il programma registrazione.pyc. Inserire i dati personali fornendo (separatamente) Numero di Matricola, Cognome e Nome. Il programma genera un file studente.txt che *non deve essere modificato manualmente*. Verificare che i dati nel file studente.txt siano corretti; in caso di errore potete rieseguire il programma registrazione.pyc.

Svolgimento degli esercizi

Leggere attentamente il testo e risolvere gli esercizi proposti.

Per ogni esercizio avete una cartella EsercN che contiene un file dal nome A_ExN.py (dove N è il numero dell'esercizio). Questo è il file che dovrete modificare con la vostra soluzione. Non create nuovi file.

Per verificare la correttezza di un esercizio **DOVETE** usare il programma TestEx.pyc (basta cliccarci sopra 2 volte) che proverà la vostra soluzione con un certo numero di casi di test. **Si noti che per la correzione verranno usati insieme di dati di test diversi.**

E' possibile consultare la documentazione ufficiale del linguaggio Python, ma **non è possibile usare libri o appunti**. In caso di problemi tecnici chiedere ai docenti o ai tecnici del laboratorio.

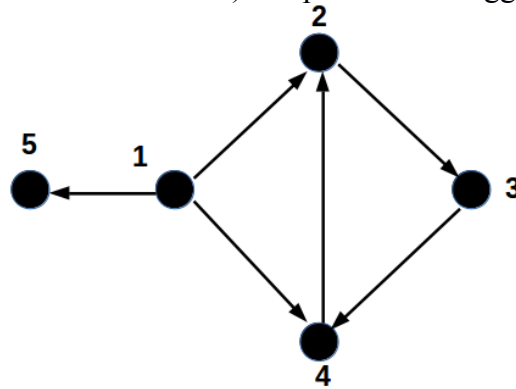
Esercizi

- **A_Ex1(s) (8 punti)** Scrivere una funzione che riceve in ingresso una stringa *s* e calcoli la lunghezza della più lunga sequenza di caratteri successivi in ordine (strettamente) crescente. Ad esempio se la *s* vale 'dbclmnoa' allora la funzione deve restituire 6, poiché la sequenza 'bclmno' in *s* è di lunghezza 6 ed è la più lunga ordinata in modo (strettamente) crescente.
- **A_Ex2(file) (8 punti)** Scrivere una funzione che prende in ingresso un file che contiene *n* interi per ogni riga separati da spazio e restituisce l'intero più piccolo e l'intero più grande. Ad esempio, se il file contiene "3 7\n55 17\n8 23\n99 12\n6 21\n" la funzione deve restituire (3,99).
- **A_Ex3(file) (8 punti)** Completare la funzione Python che riceve in ingresso una matrice di stringhe *m* e un carattere *c* e restituisce l'indice della riga *i* in cui il carattere *c* occorre più di frequente. Se il carattere *c* non è presente in alcuna riga di *m*, allora la funzione deve restituire il valore -1. Se il carattere *c* appare su più righe con la stessa frequenza massima, allora la funzione deve restituire l'indice più piccolo tra questi.

Esempio: Se la matrice *m* è $\begin{pmatrix} \text{Ciao} & \text{Bye} \\ \text{Casa} & \text{Canale} \end{pmatrix}$ e il carattere *c* è "a" allora la funzione deve

restituire 1 mentre se il carattere *c* è "z" allora la funzione deve restituire -1.

- **A_Ex4(g, u, v) (8 punti)** Implementare la funzione **A_Ex4(g, u, v)** che (tra gli altri parametri) riceve in ingresso un dizionario **g** che rappresenta un grafo *diretto*, i cui nodi hanno etichette *inter*. I parametri **u** e **v** sono due nodi del grafo. La funzione deve restituire la lista *ordinata* dei nodi che sono raggiungibili *da u ma non da v*. Si ricordi che un nodo **y** è raggiungibile a partire da un nodo **x** se esiste un cammino (diretto nel nostro caso) che parte da **x** e raggiunge **y**.



Ad esempio, se il grafo fosse quello in figura, avremmo $g = \{1: [4, 2, 5], 2: [3], 3: [4], 4: [2], 5: []\}$. Se **u** fosse il nodo 1 e **v** fosse il nodo 4, la funzione dovrebbe restituire la lista $[1, 5]$, perché i nodi 1 e 5 sono raggiungibili da 1 ma non da 4.

Si consideri che il nodo di partenza è raggiungibile per definizione.