

20° Coppa Algoritmi

Travelling Salesman Problem (TSP)

Dato un insieme di città trovare il percorso più corto che le visiti tutte senza passare due volte dalla stessa città. Le città sono rappresentate da un indice e da una coppia di coordinate nello spazio euclideo.

Lo studente deve proporre un algoritmo per risolvere il problema e testarlo su un insieme di 10 istanze date (massimo tempo di run: 3 minuti su processore singolo).

Per ogni istanza calcoliamo:

$$Error = \frac{Tour_{length} - Tour_{best}}{Tour_{best}}$$

e poi facciamo la media sulle 10 istanze. Chi ottiene il miglior risultato (errore più basso) è il vincitore della coppa (a parità di valore conta chi ha consegnato prima).

La lunghezza del tour (costo della soluzione) si calcola in base alla distanza euclidea tra le città, di conseguenza tramite la formula:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Dove (x_i, y_i) e (x_j, y_j) sono le coordinate di due città. Le coordinate sono espresse in numeri decimali ma la distanza calcolata deve essere intera (arrotondata per eccesso o difetto). Per maggiori informazioni guardare il file *ALGO_cup_2020_description.pdf*.

A fine coppa, bisogna consegnare un **RAPPORTO** (pdf + excel) e il **CODICE FINALE** e i **FILES SOLUZIONE** (*.opt.tour*) della run migliore.

Codice

Il codice deve essere scritto nel linguaggio **JAVA 8** e **NON** deve contenere alcuna chiamata a una libreria esterna se non quelle minime per il funzionamento del linguaggio stesso. L'implementazione deve essere single-core, è vietata l'implementazione di una soluzione multi-thread o che faccia uso di costrutti java multi-thread (e.g. streams).

Il codice deve essere leggibile e ben commentato per evitare fraintendimenti durante la correzione, soprattutto quando si implementano soluzioni particolarmente elaborate, in questo ultimo caso è richiesta una descrizione accurata della propria soluzione nel rapporto.

Nel caso di implementazioni di più algoritmi, specificare nel rapporto quale viene effettivamente utilizzato per raggiungere il risultato finale.

È **ASSOLUTAMENTE** vietato copiare da compagni di corso o da soluzione degli anni precedenti, il codice deve essere **PERSONALE**, la soluzione **REPLICABILE** ed il tutto **ESEGUIBILE** al di fuori del proprio laptop.

È consentito l'utilizzo di qualunque sistema operativo, ma si tenga in considerazione che la valutazione verrà effettuata su Linux, di conseguenza dovrete stare attenti che sia cross-platform.

Svolgimento

Vi verranno consegnati 10 files che corrispondono ai problemi da risolvere. I file devono essere letti così come sono dati senza modifiche nella struttura.

L'intera esecuzione del codice (dalla lettura del file fino alla scrittura della soluzione) avrà un limite di tempo fissato su **3 MINUTI**, dopo di che il processo deve essere interrotto e il risultato deve essere riportato su file con estensione *.opt.tour* che dovrà essere salvato in una directory esterna al progetto.

Consegna

Per la consegna vengono richiesti:

- Codice sorgente con la propria soluzione e i test per poterla eseguire.
- Documentazione sul lavoro svolto in PDF che deve riportare oltre alla descrizione della soluzione anche i risultati ottenuti ed eventuali seed utilizzati per raggiungere quel risultato
- Documento excel (*ALGO_cup_2020_studentname.xls*) con i risultati riportati nella tabella
- Files *.opt.tour* generati con il proprio algoritmo della run migliore

Il tutto in un archivio *.zip* con nome e cognome.

La consegna deve essere **OBBLIGATORIAMENTE** fatta via iCorsi. Codice e documentazione ricevuti in ritardo o per email non verranno valutati.

Valutazione

La valutazione verrà fatta sul un PC Desktop comune, di conseguenza è consigliato provare la propria soluzione sul sistema dove verrà valutato.

Per la valutazione il codice deve poter essere compilato con **MAVEN** e l'esecuzione dell'esperimento implementata nel file di test *SolverTest.java*. Deve essere possibile eseguire tutti i test sequenzialmente con il comando:

```
mvn test
```

ed eventualmente singoli test con:

```
mvn -Dtest=SolverTest#eil76 test
```

Per comodità vi consiglio di parametrizzare il vostro script con almeno 3 parametri:

- path file.tsp
- seed
- path file.opt.tour

I file *.out.tour* è consigliabile salvarli al di fuori dei package del vostro TSP.

Per evitare test che non partono e/o problemi nell'esecuzione del codice, vi invito a scrivere i comandi che avete usato per lanciare i test nella documentazione, di descrivere i parametri che avete usato e come usarli.

Eventuali altri parametri per uso personale sono concessi ma non saranno considerati in fase di valutazione

Maven template

Per rendere la struttura unica vi verrà fornito un template Java 8 + Maven nella quale dovrete:

- Cambiare il nome del package *ch.supsi.<cognome>* aggiungendo il vostro cognome
- Implementare i test per ogni singolo problema

All'interno del package *ch.supsi.<cognome>* avete libertà assoluta di impostare il progetto come meglio volete.

La routine di test per la risoluzione dei problemi è già impostata nel file *SolverTest.java*.