

# RELAZIONE ESAME 13/09/2024

## Strutture Dati

Nella soluzione vengono utilizzate 4 strutture dati di tipo ADT di prima classe, definite anche nel testo del problema del tema d'esame: ELENCO, DISTMATR, SEDI e SERVIZI.

### ELENCO

L'elenco delle città è definito da un vettore di stringhe città e un intero maxN che definisce la dimensione massima del vettore. Sono definite le funzioni:

- 1 ELENCOinit: inizializza l'elenco;
- 2 ELENCOfile: ottiene un elenco da file;
- 3 ELENCOgetN: ottiene il valore di maxN.

### DISTMATR

La matrice delle distanze è definita da una matrice quadrata di interi m e un intero N che definisce la quantità di righe e di colonne. Sono definite le funzioni:

- 1 DISTMATRinit: inizializza la matrice delle distanze;
- 2 DISTMATRfile: ottiene una matrice delle distanze da file a partire da una matrice delle distanze già definita;
- 3 DISTMATRgetN: ottiene la quantità di righe e colonne;
- 4 DISTMATRgetVal: ottiene il valore nella matrice in posizione (i, j).

### SEDI

Un insieme di sedi è definito da un vettore s, che indica quali città sono sedi attraverso un valore binario (se vale 0 non è una sede, altrimenti è una sede), il numero di sedi numS e la dimensione N del vettore s, definito dal numero di città. Sono definite le funzioni:

- 1 SEDlinit: inizializza l'insieme di sedi;
- 2 SEDlfile: ottiene l'insieme di sedi da file;
- 3 SEDlgetN: ottiene la dimensione del vettore s;
- 4 SEDlgetNumS: ottiene il numero di sedi;
- 5 SEDlisSede: controlla se una città definita dall'indice i è una sede.

## SERVIZI

Questa struttura dati costituisce la soluzione al problema di ottimizzazione, definita da un vettore reach che correla l'indice (che definisce la città) al valore (la sede associata alla città), il numero massimo di città maxN, il numero di sedi utilizzate e il valore della distanza media per la soluzione. Sono definite le funzioni:

- 1 `SERVIZIinit`: inizializza la struttura dati;
- 2 `SERVIZIfile`: ottiene una soluzione da file e viene utilizzata per testing;
- 3 `SERVIZIgetMaxN`: ottiene il numero massimo di sedi;
- 4 `SERVIZIgetN`: ottiene il numero effettivo di sedi;
- 5 `SERVIZIcount`: conta il numero di città associate ad una sede definita dall'indice i;
- 6 `SERVIZIgetReachVal`: ottiene il valore nel vettore reach all'indice i;
- 7 `checkPart`: controlla la validità di una soluzione.

## Funzioni

### caricaDati

La funzione usa come parametri un riferimento al file, un riferimento ad un elenco e un riferimento ad una matrice delle distanze.

Inizializza prima l'elenco da file, poi inizializza la matrice delle distanze con la dimensione dell'elenco e infine ottiene la matrice da file.

### checkSedi

La funzione usa come parametri una matrice delle distanze, i parametri MAXD e MINS e un insieme di sedi.

All'inizio controlla se è presente il numero minimo di città che non sono sedi per rendere il controllo sensato. Se è soddisfatto il controllo, allora cicla attraverso tutto il vettore s dell'insieme di sedi controllando se una città è una sede, e successivamente agisce nel seguente modo:

- 1 Se non è una sede, allora si controlla se è presente almeno un elemento che soddisfa la condizione di massima distanza. In caso contrario, ritorna falso;
- 2 Se è una sede, allora si controlla se è presente il numero minimo di città associate ad essa. In caso contrario, ritorna falso.

Se non è stato effettuato un ritorno precedente, allora ritorna vero.

## checkPart

La funzione usa come parametri una struttura dati SERVIZI, una matrice delle distanze e i parametri MAXD e MINS. Utilizza un vettore di interi check per controllare quali città sono già state controllate come sedi o raggiunte come città.

Si cicla un numero di volte pari al numero di sedi. All'interno del ciclo si ottiene il valore del primo indice nel vettore reach non ancora controllato o raggiunto. Si cicla nuovamente per aggiornare il vettore check. Se il conto delle città visitate da una determinata sede è minore di MINS, allora si ritorna falso. Se non è stato effettuato un ritorno precedente, allora ritorna vero.

## Modifiche effettuate

- 1) In ELENCO era stato mal definito il vettore "citta", creando erroneamente una stringa e non un vettore di stringhe. In SEDI è stata eliminata la componente di tipo DISTMATR, poiché non risulta utile alla risoluzione del problema;
- 2) Per alcune funzioni è stato dimenticato il tipo di ritorno;
- 3) In checkSedi è stato eliminato il secondo ciclo, inglobando le condizioni interne nel primo, poiché vengono effettuate operazioni complementari;
- 4) In checkPart sono stati rimossi i parametri SEDI \*M e int N, poiché non sono risultati utili allo scopo della funzione. La condizione di controllo `check[j] != 1` è stata modificata a `check[j] != 0`, poiché è necessario continuare il ciclo fino al primo valore utile. Sono state corrette delle dimenticanze minori relative alla sintassi.

## Commenti

All'esecuzione del codice vengono effettuati 4 test: 2 per sedi e 2 per servizi.

I test relativi alle sedi controllano che la funzione checkSedi funzioni correttamente, evidenziando l'output atteso e l'output effettivo del codice a partire dal file di test "testSedi.txt".

I test relativi ai servizi controllano che la funzione checkPart funzioni correttamente, evidenziando l'output atteso e l'output effettivo del codice a partire da due file di test "testServizi1.txt" e "testServizi2.txt".