

Strutture dati adottate:

Tipo ELENCO: Struct contenente vettore di stringhe per l'elenco delle città

Tipo DISTMATR: Struct contenente matrice di interi per la matrice delle distanze

Tipo SEDI: Struct contenente un vettore di M interi rappresentanti l'indice delle città sede di p.s.

Tipo SERVIZI: struct contenente un assegnazione di città senza p.s. a città sedi di p.s.. Contiene:

- `dist_media`: distanza media tra le città sedi di p.s. e le città senza p.s.
- `città_no_ps`: vettore di N-M interi. indici delle città senza p.s. (complementare a di quello di sedi)
- `sol`: vettore di N-M interi. l'i-esimo elemento è l'indice (nell'elenco generale) della città a cui l'i-esima città senza p.s. (i-esima nel vettore `città_no_ps`) è stata assegnata
- `servite`: vettore di M interi. l'i-esimo elemento è il numero di città senza p.s. servite dalla i-esima città sede di p.s. (i-esima nel vettore in sedi)

Nota: il vettore `servite` si può calcolare a partire dal vettore `sol`, ma è comodo averlo già calcolato per fare i controlli

Strategie risolutive:

caricaDATI: a questa funzione viene passato il puntatore al file di tipo FILE* e le due strutture dati ELENCO e DISTMATR passate by reference

checkSedi: fa un controllo preliminare sulla validità della scelta di sedi di p.s.. Controlla che:

- per ognuna delle altre N-M città esista almeno una città sede di p.s. a distanza minore di MAXD
- ogni città sede di p.s. serva almeno MINS città senza p.s. (a distanza < di MAXD)

bestPart: wrapper a una funzione ricorsiva `bestPart_r` che prova tutti i modi possibili di assegnare le città senza pronto soccorso a quelle con pronto soccorso. Lo fa riempiendo un vettore `sol` (contenuto all'interno della struttura dati SERVIZI) con gli indici delle città sede di pronto soccorso (gli indici dell'elenco generale). Molti dei rami della ricorsione vengono tagliati dalla funzione

prunePart, che agisce eseguendo questi due controlli:

- Se a una città sede di p.s. sono assegnate più di MAXS città, esegui il pruning. MAXS è calcolato come: (numero di città senza p.s.) - MINS * (numero di città sede di p.s. - 1). Infatti, se a una città sede di p.s. vengono assegnate più di MAXS città senza p.s., non rimangono abbastanza città senza p.s. per le altre città sede di p.s.
- A una città sede di p.s. viene assegnata una città troppo lontana

Quando il vettore è pieno, la funzione **checkPart** controlla se la soluzione è valida nel seguente modo:

- A ogni città sede di p.s. devono essere assegnate almeno MINS città senza p.s.
- Il controllo delle distanze (<MAXD) non deve essere fatto in quanto è già fatto da `prunePart`

Se la soluzione corrente è valida e ha una distanza media minore della soluzione migliore, la soluzione corrente diventa la soluzione migliore

Differenze tra esame e relazione:

- **Struttura SERVIZI e link:** durante l'esame non mi è venuto in mente di usare un vettore `sol` di interi nella funzione ricorsiva, all'inizio ho pensato di dotare SERVIZI di M liste, che avrei usato per aggiungere/togliere le città senza p.s. nella funzione ricorsiva. Perciò avevo anche introdotto la struttura dati per le liste contenenti interi con le funzioni `link addToList(link l, int i);` e `link removeFromList(link l, int i);`. Avendo rimosso queste funzionalità nella revisione ho cambiato anche la funzione `SERVIZI serviziIniz;`
- **Funzione SERVIZI sol from appartenenza c n ps(int* appartenenza c n ps);**: mi scuso innanzitutto del nome incomprensibile, a questo punto dell'esame mi era venuto in mente di usare un vettore di interi `sol` come descritto sopra (che purtroppo ho scelto di chiamare `appartenenza_c_n_ps`), ma non mi era venuto in mente di incorporarlo all'interno di SERVIZI. Perciò questa funzione serviva a creare le liste della struttura SERVIZI a partire dal vettore `sol`, e sarebbe stata chiamata prima di chiamare `prunePart` e `checkPart`.
- **Funzione checkPart:** durante l'esame non ho avuto tempo di scriverla
- **Funzione prunePart:** ho corretto la formula per MAXS e ho aggiunto il controllo di MAXD
- **Errori e dimenticanze varie:** come per esempio nella funzione `caricaDATI` ho trattato ELENCO e DISTMATR come se fossero passati by value, mentre li avevo passati by reference; nelle funzioni `checkSedi` e `ellnVector` ho scritto `if (x == y)` anziché `if (x == y);`; avevo definito il parametro `distanza media` all'interno di SERVIZI come `int` e non `float`;