TSP

In questa documentazione c’è la descrizione della classe TSP programmata per essere eseguita esclusivamente nel compito di ricerca operativa.

La classe tsp ha un vettore di dimensione 10, con in più un albero da allocare, che sarà di 3 piani.

Il vettore è di 10 dimensioni, perché sappiamo con certezza che i nodi sono 5 e che il tsp è simmetrico, per cui ci bastano 10 celle.

L’albero binario ci servirà per il tsp e per stampare i risultati.

La classe tsp ha molte funzioni di utilità, ma poche funzioni pubbliche, tra cui:

* Costruttore tsp(), che alloca in memoria un vettore di 10 celle, i cui valori saranno stabiliti dall’utente e ogni cella corrisponde alla combinazione (i,j) con riga i e colonna j, disposta in ordine lessicografico Xij. Ci sarà possibilità di controllare con una stampa diretta se la digitazione sia stata corretta e verrà chiesto di confermare con 1 la combinazione o di ripeterla con 0 e correggerla, indicando il numero di errori, e poi per ciascun di essi di indicare rispettivamente il valore corretto e la posizione in cui si sarebbe verificato l’errore.

Inoltre, viene allocato in memoria anche il  
L’albero è vuoto vettore w di dimensione 15, aventi Vi e Vs nulli, e aventi stati di vuoto, tagliato e hamiltoniano settati a false. Le celle sono allocate secondo la convenzione vista nel corso di ricerca operativa che corrispe all’ordine dal nodo in alto a quello in basso e da sinistra verso destra, come previsto da convenzione.

* Distruttore ~tsp, che distrugge l’albero
* Branch\_Bound, funzione che si comporta come segue e con l’ordine così descritto:

1. Viene chiesto di digitare rispettivamente il nodo per il k\_albero e il nodo più vicino a k;
2. Poi la funzione chiama la funzione che calcola il k\_albero, indicando i nodi scelti e in cui non è collegato il nodo k scelto(l’albero di copertura), e poi gli archi in cui è coinvolto il nodo k. Viene poi stampato la valutazione inferiore (Vi) che è la somma dei pesi degli archi;
3. Poi viene chiamata la funzione che stampa gli archi scelti in ordine secondo quanto previsto dal nodo più vicino a k e la valutazione superiore(Vs);
4. Poi chiede di digitare per 3 volte di fila i nodi che partono da i e arrivano a j (i,j), cioè secondo le convenzioni del corso di ricerca operativa dei nodi (i,j), per poter allocare l’albero completo;
5. Chiama la funzione branch, una funzione di utilità che svolgerà i conti necessari per scoprire i valori di Vi, se il ciclo è hamiltoniano o se la soluzione del nodo è vuoto e le informazioni vengono aggiornate nel vettore w., seguendo l’ordine dall’alto verso il basso e da sinistra verso destra;
6. Chiama la funzione aggiorna\_Vs, che aggiorna le Vs correttamente a partire del nodo più sinistra all’altezza h in cui si verifica un ciclo hamiltoniano;
7. Chiama la funzione taglia\_rami che taglia i rami, nel caso in cui Vi >= Vs oppure se si presenta un ciclo hamiltoniano;
8. Chiama stampa\_risultato che stampa il risultato come prevista dalla convenzione vista a lezione, per una leggera eccezione: vengono aggiornati tutti i nodi a partire da h sempre da sinistra, risultando l’algoritmo comunque corretto(il prof conferma la sua correttezza formale)

Branch

La funzione Branch è la funzione che calcola la Vi. Come funziona? Viene chiamata la funzione branch Dato l’albero binario e il vettore w allocati, come descritto nel paragrafo precedente, crea un vettore fittizio che ci terrà in mente quali nodi includere ed escludere nell’albero fa una visita anticipata dell’albero e:

* Nel caso in cui non fossimo al piano 0, si può seguire le indicazioni fornite dall’albero stesso che ci indicherà quale posizione della cella è da includere o se da escludere e poi includerà o escluderà quel nodo
* Durante l’inclusione del nodo ci si assicura che: che non vengano inclusi nodi che generino un ciclo disgiunto o 3 archi nel nodo k. In caso contrario, viene stampato “Vuoto” e si esce dalla chiamata
* Durante l’esclusione del nodo ci si assicura che tutti gli archi da collegare ad un certo nodo riga o colonna dell’albero abbia almeno 2 nodi collegabili. Se a quel nodo vengono vietati i collegamenti con altri 3 nodi, la soluzione sarà vuota e per cui verrà stampato “Vuoto” e si esce dalla chiamata
* Calcola il k\_albero del nodo corrente, considerando eventuali nodi tolti da aggiornare nel vettore, o già scelti, anch’essi da aggiornare nel vettore della classe
* Controlla se il ciclo è hamiltoniano.
* Ogni salto è una chiamata ricorsiva e vengono gestiti gli aggiornamenti. Nelle funzioni di utilità gli input sono sanitizzati e provvedono a chiudere forzatamente l’esclusione con il codice di uscita 1.
* Ogni informazione ricavata viene salvato in w, sapendo che ogni figlio del padre è padre \* 2, incrementato di 1 se è il figlio sinistro, e +2 se il figlio è destro.

Trova\_Vs

Questa funzione aggiorna la Vs correttamente, assumendo che ogni cella da sinistra verso destra corrisponda alla posizione del nodo che parte dall’alto verso il basso e da sinistra verso destra.

Tenendo a mente la Vs trovata a partire dal padre si aggiorna e si assegna ad ogni nodo, a partire dall’altezza h, le Vs e salvate le informazioni nel vettore w.

La convenzione scelta funziona così:

- se la cella ha lo stato false sia a hamiltoniana, che a vuoto, gli si aggiorna la Vs più piccola trovata

- se la cella ha lo stato hamiltoniano a true e vuoto a false, ma la Vs aggiornata è più piccola della Vi, allora la si aggiorna comunque quella Vs

- se la cella ha lo stato hamiltoniano a true e vuoto a false, ma quella Vs è la più piccola, allora non c’è nessun problema e non la si aggiorna

Taglia\_Rami

Questa funzione, se trova nodi da tagliare, si procede a tagliare ricorsivamente i suoi figli, limitandosi a tenere lo stato tagliato a true ed ignora i nodi già tagliati.

Stampa\_risultato

Questa funzione stampa il risultato seguendo le seguenti regole:

1. Se il ciclo non è hamiltoniano oppure è ciclo hamiltoniano, ma risulta avere la Vs più grande della Vs minima a partire da quella altezza, viene scritto il formato Pi,j: [Vi,Vs] = [Vi-valore,Vs-valore] e si va a capo
2. Se il ciclo è hamiltoniano oppure Vs = Vi, allora viene stampato il formato Pi,j : Vi = Vs = Vi-valore e si va a capo
3. I nodi tagliati non vengono stampati
4. Per ciascun piano superato si va a capo una volta in più