



Commento al Laboratorio n. 4

Esercizio n.1: Massimo Comun Divisore

Si ricordi che nel calcolo del massimo comun divisore si assume che $a > b$. Per tener conto di tale condizione, se non soddisfatta, si scambiano a e b . La condizione di terminazione si raggiunge o quando a è uguale a b oppure quando b è 0. In entrambi i casi si ritorna a.

Si identificano 4 casi:

1. a e b sono entrambi pari: 2 è certamente un fattore primo, quindi il massimo comun divisore di a e b è il doppio del massimo comun divisore di $a/2$ e $b/2$
2. a è dispari e b è pari: il fattore primo 2 di b non compare certamente nel massimo comun divisore di a e b , che quindi è il massimo comun divisore di a e $b/2$
3. dualmente a è pari e b è dispari: il fattore primo 2 di a non compare certamente nel massimo comun divisore di a e b , che quindi è il massimo comun divisore di $a/2$ e b
4. a e b sono entrambi dispari: secondo l'algoritmo di Euclide basato sulla sottrazione il massimo comun divisore di a e b è il massimo comun divisore di $a-b$ e b , ma $a-b$ è certamente pari, quindi è applicabile il caso 3. Il massimo comun divisore di a e b è quindi il massimo comun divisore di $(a-b)/2$ e b .

Esercizio n.2: Elemento maggioritario

La soluzione verrà pubblicata dopo il termine di caricamento degli esercizi valutati.

Esercizio n.3: Valutazione di espressioni regolari

La soluzione verrà pubblicata dopo il termine di caricamento degli esercizi valutati.

Esercizio n.4: Azienda di trasporti – ordinamento

Struttura dati: la `struct` per la tabella del Lab. 2 (tipo `tabella_t`) viene estesa a comprendere un campo per il criterio di ordinamento corrente, il cui valore è selezionato in un tipo `chiaveOrdinamento` definito per enumerazione (NESSUNO, DATA, CODICE, PARTENZA, ARRIVO). Si tratta a tutti gli effetti di una `struct` wrapper (cfr. *Puntatori e strutture dati dinamiche 5.3*) che contiene informazioni non necessariamente omogenee tra loro.

Si osservi la tabella utilizzata nel Lab. 2 era una `struct`, uno dei cui campi era un vettore. La `struct` veniva passata alle diverse funzioni *by value*, quindi copiata, con evidente costo in termini sia di tempo, sia di memoria. Si propongono, in aggiunta alla soluzione-base, 2 soluzioni rese più efficienti dai passaggi *by reference*:

1. si disaccoppiano le informazioni raggruppate nella `struct` (vettore, sua dimensione e criterio di ordinamento), passandole quando necessario separatamente come parametri. Poiché il passaggio come parametro di un vettore è sostanzialmente *by reference* si evita la ricopiatura
2. la `struct` viene passata non *by value*, bensì emulando il passaggio *by reference* ricopiando il puntatore alla `struct` stessa.

Menu: invece di utilizzare un tipo definito per enumerazione per i comandi, nella soluzione proposta si utilizzano esplicitamente gli interi come indici di un vettore di stringhe che li contiene.

Ordinamento: si utilizza l'insertion sort per realizzare un ordinamento stabile. La funzione di confronto `confrontaVoci` tiene conto della chiave di ordinamento richiesta per identificare i campi da confrontare e le funzioni di confronto (`comparaData`, `comparaOra` o `strcmp`).



Ricerca: per decidere se applicare una ricerca lineare o dicotomica si testa secondo quale chiave è eventualmente ordinato il vettore. Anche se non richiesto dalle specifiche, si realizza una funzione di ricerca per codice, per la quale sia la ricerca lineare, sia quella dicotomica sono standard.

La ricerca parziale è fatta per prefissi della stringa, quindi a partire dal primo carattere, mediante la funzione `strncmp`. Essa richiede di conoscere il numero di caratteri per il confronto. Al tal scopo si ricorre alla funzione `strlen`.

La ricerca in base a una stazione di partenza può trovare più di un risultato. Se è richiesto di elencarli tutti:

- nel caso di utilizzo della ricerca dicotomica, si trova una qualunque delle stazioni di partenza corrispondenti, dalla quale (se trovata) si reperiscono le altre mediante determinazione di un intervallo: se presenti, altre stazioni di partenza corrispondenti al criterio sono adiacenti, nel vettore ordinato, alla stazione di partenza trovata
- nel caso di ricerca lineare, si itera su tutto il vettore, stampando il risultato non appena trovato la stazione di partenza o parte di essa.

Esercizio n.5: Azienda di trasporti - multiordinamento

La soluzione verrà pubblicata dopo il termine di caricamento degli esercizi valutati.