Algoritmi e strutture dati

Docente: Violetta Lonati

Prova di laboratorio - appello del 16 giugno 2022

Note importanti

- Si leggano attentamente i testi degli esercizi e le indicazioni su come svolgerli. Se ci sono dubbi sul significato delle richieste, è opportuno chiedere chiarimenti!
- Si leggano attentamente anche le indicazioni su come preparare le risposte. Per ogni esercizio è richiesto di preparare un file: in alcuni casi si tratta di un file di testo, in altri casi di un programma in C. Per ogni esercizio viene indicato il nome con cui salvare il file; è importante rispettare questa indicazione.
- Nella prima riga di tutti i file consegnati è necessario **scrivere nome, cognome e matricola**.
- Dopo essersi autenticati, si carichino sul sito `upload.di.unimi.it` i file contenenti le risposte.

I nomi dei file consegnati devono essere i seguenti:
`es1-matrice.c`, `es2-misteriosa.txt`, `es3-quadretti.txt`, `es4-quadretti.c`.

1 Matrice - esercizio filtro

Si scriva un programma che:

- legga da standard input una matrice di bit (si veda sotto per il formato dell'input);
- memorizzi la matrice di bit allocando *dinamicamente* lo spazio opportuno;
- calcoli e stampi il numero di bit pari a 1 contenuti nella matrice;
- interpreti ogni riga della matrice come numero binario, lo converta in decimale e lo stampi.

Formato di input Il programma deve leggere due numeri n e m che indicano il numero di righe e di colonne della griglia, seguiti da n righe di m bit 0 oppure 1.

Esempio di esecuzione Se il programma riceve da standard input i numeri:

```
3 5
00100
01010
00000
```

allora il programma stampa in input

```
3 4 10 0
```

dove il 3 indica il numero di caselle nere, il 4 indica la conversione in decimale di `00100`, il 10 indica la conversione in decimale di `01010`, lo 0 indica la conversione in decimale di `00000`.

Note per la consegna. Si scriva il programma in un file con nome `es1-matrice.c`.

2 Struttura dati misteriosa

Considera la struttura dati riportata qui sotto e rispondi ai punti nelle pagine successive.

```
1  // Item is a type defined elsewhere
2
3  typedef struct {
4    Item *c;
5    int lo;
6    int hi;
7    int len;
8    int n;
9  } Ds;
10
11
12 Ds *ds_new() {
13   Ds *new = malloc(sizeof(Ds));
14   new -> len = 1;
15   new -> n = new -> lo = new -> hi = 0;
16   new -> c = malloc(new -> len * sizeof(Item));
17   return new;
18 }
19
20 void ds_rebuild(Ds *y) {
21   Item *tmp = malloc(2 * y -> len * sizeof(Item));
22   for (int i = 0; i < y -> n; i++ )
23       tmp[i] = y -> c[(i + y -> lo) % y -> len];
24   free(y -> c);
25   y -> c = tmp;
26   y -> lo = 0;
27   y -> hi = y -> n;
28   y -> len *= 2;
29 }
30
31 void ds_insert(Ds *y, Item in) {
32   y -> c[y -> hi] = in;
33   y -> hi += 1;
34   if ( y -> hi == y -> len)
35     y -> hi = 0;
36   y -> n += 1;
37   if (y -> n == y -> len)
38     ds_rebuild(y);
39 }
40
41 Item ds_remove(Ds *y) {
42 // assumes y is not empty
43   Item out = y -> c[y -> lo];
44   y -> lo += 1;
45   if ( y -> lo == y -> len)
46     y -> lo = 0;
47   y -> n -= 1;
48   return out;
49 }
```

1. Di che tipo di struttura dati si tratta?

   a) Coda di priorità
   b) Pila
   c) Ricerca binaria
   d) Albero
   e) Divide et impera
   f) Set
   g) Union & find
   h) Lista concatenata
   i) Coda
   j) Grafo

2. Descrivi con parole tue come è implementata la struttura dati. Non spiegare il codice riga per riga, ma cerca di spiegare come è implementata a livello generale. Sii più preciso e sintetico possibile, ma non più breve del necessario; se riempi più di mezza pagina, sei al livello di astrazione sbagliato.

3. Scrivi una funzione `int ds_size(ds *y)` che restituisce il numero degli elementi nella struttura.

4. Scrivi una funzione `void ds_print(ds *y)` che stampa la struttura.

5. Quali di queste relazioni tra i membri della struttura restano sempre vere, dopo una chiamata di `ds_new` seguita da una sequenza di chiamate delle funzioni `ds_insert` e `ds_remove`? NB: non è importante se la relazione cambia durante l'esecuzione delle funzioni, importa che sia vera al termine delle funzioni.

   Per ciascuna delle opzioni, stabilisci se la relazione resta vera oppure no. Nel primo caso spiega perché, nel secondo caso indica una situazione in cui questo non si verifica:

   a) $\texttt{n} < \texttt{len}$

   b) $\texttt{lo} \leq \texttt{hi}$

   c) $\texttt{hi} < \texttt{n}$

   d) $\texttt{hi} = \texttt{n}$

6. Si assuma che `Item` sia definito come `int`, che `c` contenga i valori $\{3,8,4,1\}$, `lo` valga 3, `hi` valga 2 e `n` valga 2. Si tratta di una situazione che può verificarsi dopo una sequenza di `ds_insert` and `ds_remove`? Se sì, fornisci una tale sequenza, altrimenti indica tutti i motivi per cui non si può verificare.

7. Disegna la struttura dati e i suoi membri (incluso il vettore `c` e i valori di `hi`, `lo`, `n` e `l`) dopo l'esecuzione delle seguenti operazioni:

```
Ds *y = ds_new();
ds_insert(y,1);
ds_remove(y);
ds_insert(y,2);
ds_remove(y);
ds_insert(y,3);
```

   Indica inoltre quante volte è stata eseguita la funzione `ds_rebuild` e in quali punti del codice.

8. Considera questa sequenza (parzialmente specificata) di operazioni:

```
Ds *y = ds_new();
// sequenza non specificata di ds_insert/ds_remove su y
ds_insert(y,1);
ds_insert(y,2);
ds_insert(y,3);
```

   Quale può essere il contenuto di `c` dopo l'esecuzione? Per ciascuna delle opzioni proposte, stabilisci se è possibile oppure no, giustificando ogni risposta.

   a) | | 1 | 2 | 3 |

   b) | | 1 | 2 | 3 | | |

   c) | 2 | 3 | | 1 |

   d) | 2 | 3 | | | | | 1 |

   e) | 1 | 2 | | 3 |

9. È vero o falso che la struttura dati usa spazio lineare rispetto al numero degli elementi che contiene? Giustifica la risposta.

10. Sia T il numero di elementi contenuti nella struttura dati. Nel caso peggiore, quanti accessi agli elementi di un array sono effettuati da una singola chiamata di `ds_remove`? Scegli la risposta corretta e motivala (assumiamo che il compilatore non effettui nessuna ottimizzazione, e che quindi ogni accesso scritto nel codice verrà eseguito effettivamente):

    a) $\sim 4T$

    b) 1

    c) $\sim 2T$

    d) 7

11. Nel caso peggiore, quanti accessi agli elementi di un array sono effettuati dalla chiamata più onerosa tra `ds_insert` e `ds_remove`? Giustifica la risposta.

    a) $\sim 4T$

    b) 1

    c) $\sim 2T$

    d) 7

Note per la consegna. Si scrivano le risposte alle domande in un file di testo con nome `es2-misteriosa.txt`.

3 Quadretti neri - progettazione

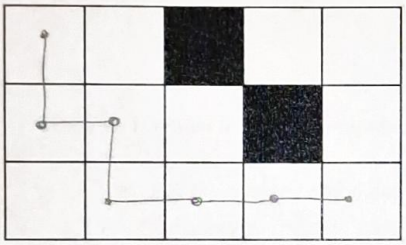
Si consideri una griglia rettangolare formata da n righe e m colonne. Le caselle della griglia possono essere nere o bianche. Chiamiamo *percorso* della griglia una qualunque sequenza di caselle $c_0, c_1, \ldots c_k$ della griglia che abbia queste proprietà:

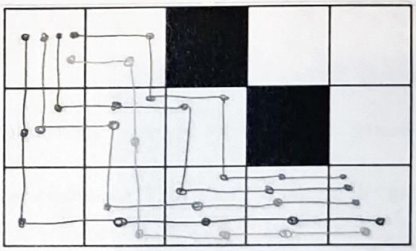
- le caselle della sequenza sono tutte bianche;
- per ogni $i = 1, \ldots k$, la casella c_i è subito alla destra di c_{i-1} oppure subito sotto a c_{i-1}.

Diciamo che il percorso $c_0, c_1, \ldots c_k$ *parte* dalla casella c_0 e *arriva* nella casella c_k. Inoltre chiamiamo *completo* un percorso che parte dalla casella in alto a sinistra e arriva nella casella in basso a destra.

Vogliamo calcolare, data una griglia qualsiasi, quanti sono i percorsi completi.

Esempio Nella figura a sinistra è mostrata una griglia ed un percorso completo. Nella figura a destra sono mostrati tutti e 5 i percorsi completi della griglia (un colore diverso per ogni percorso)





Per calcolare il numero di percorsi completi, considerate la funzione f così definita: data una casella c qualsiasi, $f(c)$ è pari al numero di percorsi che partono nella casella c e finiscono nella casella in basso a destra.

Completate le seguenti frasi / rispondete alle seguenti domande (giustificando le risposte):

1. Se c è una casella nera, allora $f(c) = \ldots$
2. Se c è la casella in basso a destra ed è bianca, allora $f(c) = \ldots$
3. Se c è una casella qualsiasi dell'ultima riga, si può scrivere una relazione di ricorrenza che fa riferimento alla casella a destra di c:
   $f(c) = \ldots$
4. Se invece c è una casella qualsiasi dell'ultima colonna come si può scrivere $f(c)$?
5. Se c è una casella bianca non di bordo, si può scrivere una relazione di ricorrenza che fa riferimento alla casella sotto e alla casella a destra di c:
   $f(c) = \ldots$
6. Per quale casella c si ha che $f(c)$ coincide il numero di cammini completi?

Progettazione Sfruttando le proprietà della funzione f sopra analizzata, progettate e scrivete un algoritmo che calcoli tutti i percorsi completi di una griglia data.

Note per la consegna. Si scriva lo svolgimento dell'esercizio in un file di testo con nome `es3-quadretti.txt`.

4 Quadretti neri - implementazione

Si scriva un programma che:

- legga da standard input una matrice di bit che descrive una griglia di quadretti bianchi e neri (0 per le caselle bianche e 1 per le nere);
- allochi *dinamicamente* una matrice che contenga la descrizione della griglia;
- calcoli e stampi il numero di percorsi completi nella griglia.

Per il formato di input si faccia riferimento a quanto specificato nell'esercizio 1.

Esempio di esecuzione Se il programma riceve da standard input

```
3 5
00100
00010
00000
```

che rappresenta la griglia della figura qui sopra, allora il programma stampa in output il numero 5.

Note per la consegna. Si scriva il programma in un file con nome `es4-quadretti.c`.