

Studente Filippo Callegari

Attenzione: una volta letto il presente testo è *obbligatorio* consegnare alla scadenza quanto elaborato, indipendentemente dal fatto che lo si consideri adeguato o meno.

Entro il termine stabilito si *deve* inviare via email dentro un unico file compresso, il cui nome sia “ProgettoLC parte2a Filippo Callegari”, tutti (e soli) i sorgenti sviluppati, in modo che sia possibile verificare il funzionamento di quanto realizzato. In tale file compresso *non* devono comparire file oggetto e/o binari o qualsiasi altro file che viene generato a partire dai sorgenti. Sempre in detto file va inoltre inclusa una relazione in formato PDF dal nome “ProgettoLC parte2a Filippo Callegari Relazione”. La relazione può contenere immagini passate a scanner di grafici o figure fatte a mano.

Si richiede:

- una descrizione dettagliata di tutte le tecniche **non**-standard impiegate (le tecniche standard imparate a lezione non vanno descritte).
- una descrizione delle assunzioni fatte riguardo alla specifica, sia relativamente a scelte non previste espressamente dalla specifica stessa, che a scelte in contrasto a quanto previsto (con relative motivazioni).

Non è assolutamente utile perdere tempo per includere nella relazione il testo dei vari esercizi o un suo riassunto o una qualsiasi rielaborazione, incluso descrizioni del problema da risolvere. Ci si deve concentrare solo sulla descrizione della soluzione e delle eventuali variazioni rispetto a quanto richiesto.

- una descrizione sintetica generale della soluzione realizzata.
- di aggiungere un file di testo con alcune query di test (da poter eseguire in un ambiente interattivo come GHCi). Oppure fornire un Makefile.

Esercizio

Si implementino immagini di $2^n \times 2^n$ pixels utilizzando opportunamente QuadTrees (alberi i cui nodi interni hanno 4 figli) ottenuti nel seguente modo.

- Se tutti i pixel dell'immagine sono uguali allora si utilizza una foglia del QuadTree contenente il “colore” dei pixel.
- Altrimenti si suddivide l'immagine in 4 blocchi di $2^{n-1} \times 2^{n-1}$ pixel e si costruisce un nodo interno i cui 4 figli sono i QuadTrees ottenuti dalla codifica dei suddetti blocchi, nell'ordine superiore-sinistro, superiore-destro, inferiore-sinistro e inferiore-destro.

QuadTrees in cui un nodo abbia come figlie 4 foglie uguali *non* sono validi.

Utilizzando il data-type

```
data QT a = C a | Q (QT a) (QT a) (QT a) (QT a)
```

si scriva una funzione `boundPicture` che dati due “colori” c_{min} , c_{max} ed un QuadTree q costruisce il QuadTree dell'immagine che si ottiene andando a limitare inferiormente i colori dell'immagine codificata da q con c_{min} e superiormente con c_{max} . Ad esempio

```
let d = C 20; u = C 10; q = Q d u u u
in boundPicture 4 15 (Q q (C -10) (C 3) q)
```

restituisce `Q (Q (C 15) (C 10) (C 10) (C 10)) (C 4) (C 4) (Q (C 15) (C 10) (C 10) (C 10))`.