

(More) Abstract Data Types

Lecture 15

Altre strutture dati

Chapter 7 of HR book

Oggi

- Strategia di ricerca su grafi
- Stacks
- Queues
- Esercizi sui moduli

Queues

- Struttura FIFO che richiede accesso avanti e retro
- `module Queue`
`type Queue<'a>`
`val empty: Queue<'a>`
`val put : 'a -> Queue<'a> -> Queue<'a>`
`val get : Queue<'a> -> 'a * Queue<'a>`
`exception EmptyQueue`

Queue come liste

Type Queue<'a> = Q of 'a list

- Put è lineare (append a fine lista)
- Get è costante (cons)

- Proviamo una lista *snoc*

Queue<'a> = Emp | Sn of 'a list* 'a

- Situazione simmetrica

- Put è costante
- Get è lineare

Queue come **coppia** di liste

Queue<'a> = Q of 'a list * 'a list

- La prima è il *front* della coda
- La seconda il retro (*rear*) della coda, ma in **ordine inverso**
- Put è costante – cons sul *rear*
- Get è costante – cons sul *front*
- Se il *front* è vuoto, lo riempio rigirando *rear*
- La complessità ammortizzata di una sequenza di n operazione è $O(n)$ vs $O(n^2)$ delle alternative