

ESERCIZIO 5 (Heap binomiali)

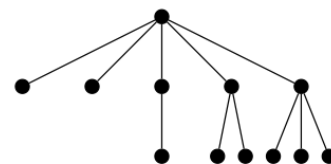
- (a) Si definiscano gli *alberi binomiali* e si enuncino le loro principali proprietà, dimostrandole adeguatamente.
- (b) Si definiscano gli *heap binomiali* e si fornisca una maggiorazione al grado massimo di un ~~un~~ nodo in uno heap binomiale contenente n nodi.

ESERCIZIO 5 (Heap di Fibonacci)

- (a) Si definiscano gli alberi binomiali non ordinati, enunciando e dimostrando le loro più importanti proprietà.
- (b) Si indichino le operazioni supportate dagli heap di Fibonacci e con quale complessità.
- (c) Sia x un nodo di grado k in un heap di Fibonacci e siano y_1, \dots, y_k i figli di x nell'ordine in cui sono stati innestati in x . Quale limitazione inferiore è possibile dare per $\text{degree}(y_i)$? Perché?

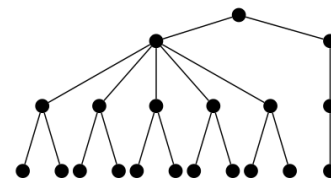
ESERCIZIO 3

- (a) Si enunci e si dimostri un lemma che fornisca una minorazione dei gradi dei figli di ciascun nodo in un heap di Fibonacci.
- (b) Si stabilisca se possa esistere un heap di Fibonacci avente la struttura dell'albero a lato.



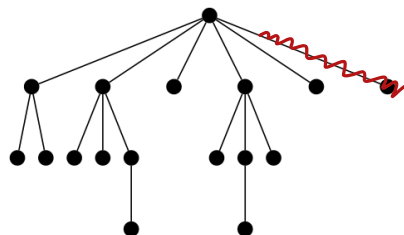
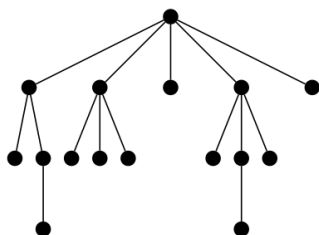
ESERCIZIO 3

- (a) Si enunci e si dimostri un lemma che fornisca una minorazione dei gradi dei figli di ciascun nodo in un heap di Fibonacci.
- (b) Si stabilisca se possa esistere un heap di Fibonacci avente la struttura dell'albero a lato.



ESERCIZIO 4 (Heap di Fibonacci)

- (a) Si enunci e si dimostri un lemma che fornisca una minorazione dei gradi dei figli di ciascun nodo in un heap di Fibonacci.
- (b) Si stabilisca, giustificando le risposte, se possano esistere degli heap di Fibonacci aventi la struttura dei seguenti alberi:

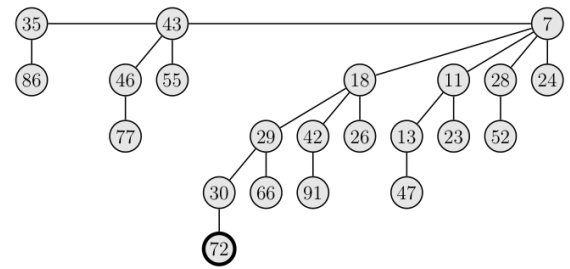


ESERCIZIO 3 (Heap binomiali)

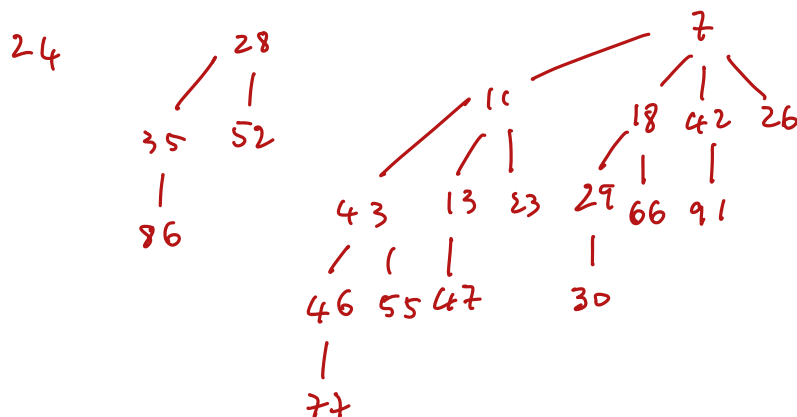
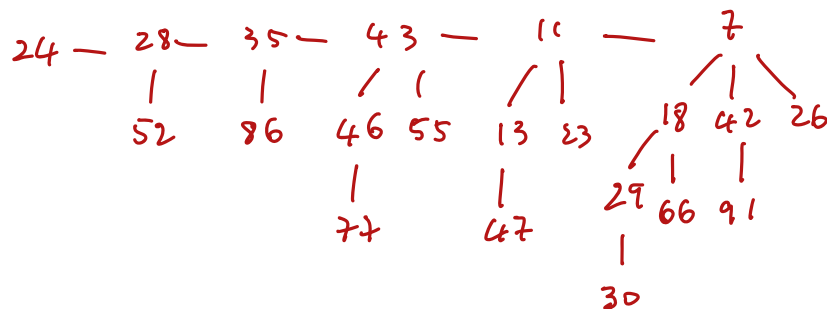
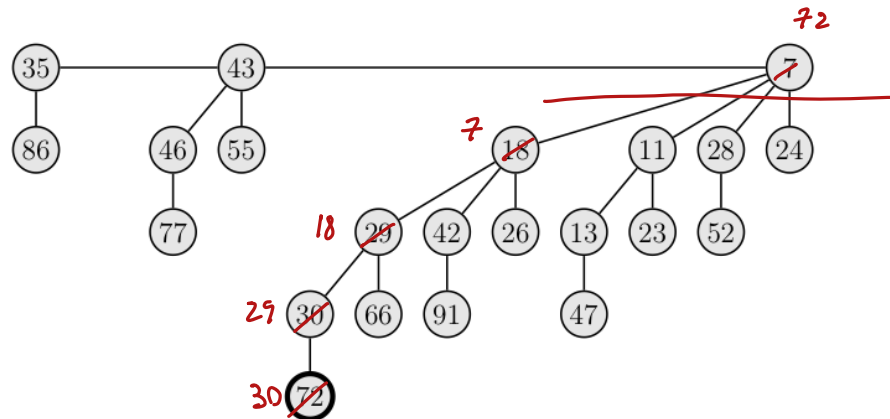
(a) Si definiscano gli *heap binomiali* e si descrivano le operazioni DECREASEKEY, EXTRACTMIN, DELETE e INSERT. Quindi si cancelli il nodo evidenziato (contenente la chiave 72) dall'heap binomiale a lato e poi si inseriscano in successione le chiavi 10, 30 e 5.

(b) Si determinino un limite superiore ed un limite inferiore per il numero di alberi binomiali in un heap binomiale con n chiavi. ≥ 1

(c) Nel caso degli heap binomiali, è richiesto che gli alberi binomiali nella lista delle radici siano *ordinati* per grado. Perché?

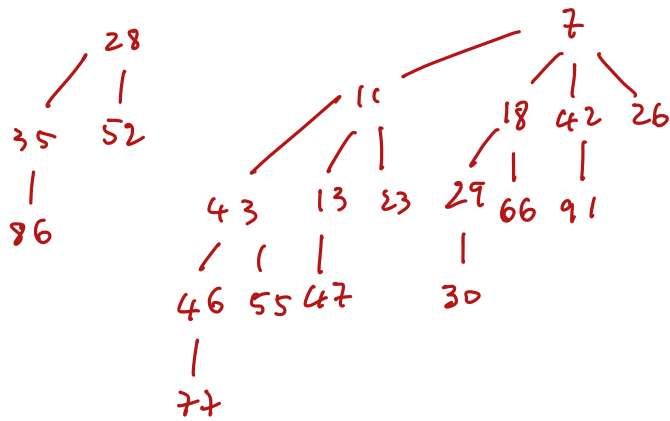


DELETE 72

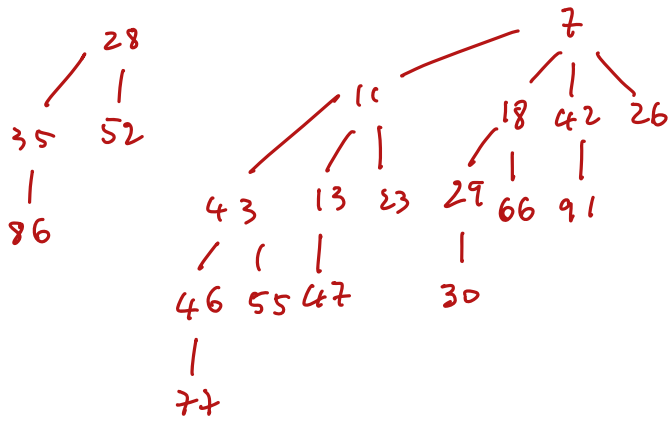


INSERT (10)

10 24



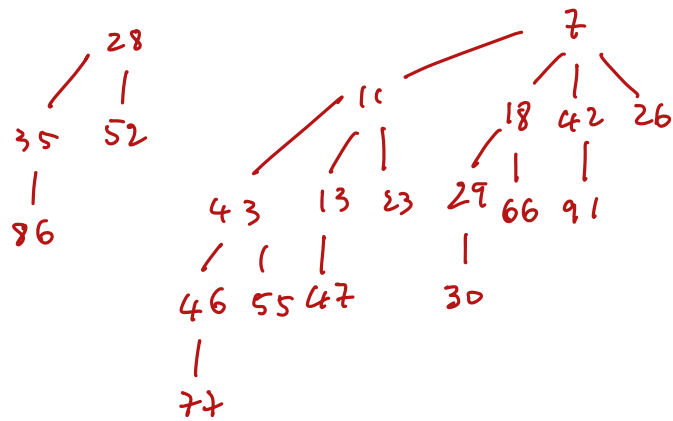
10
|
24



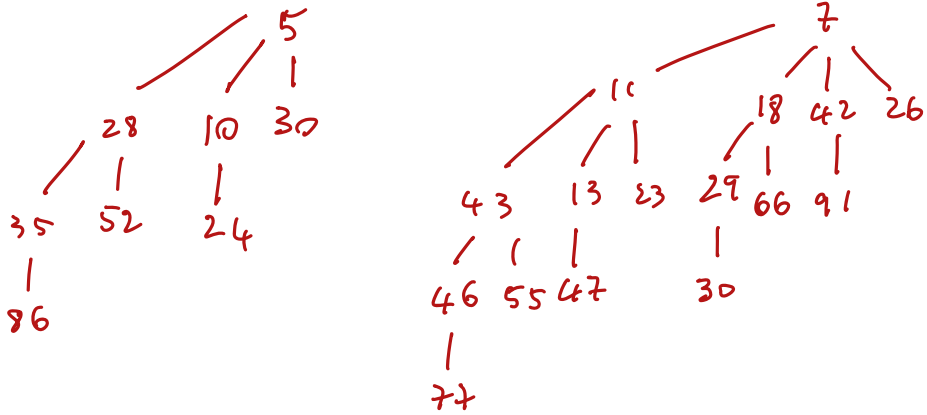
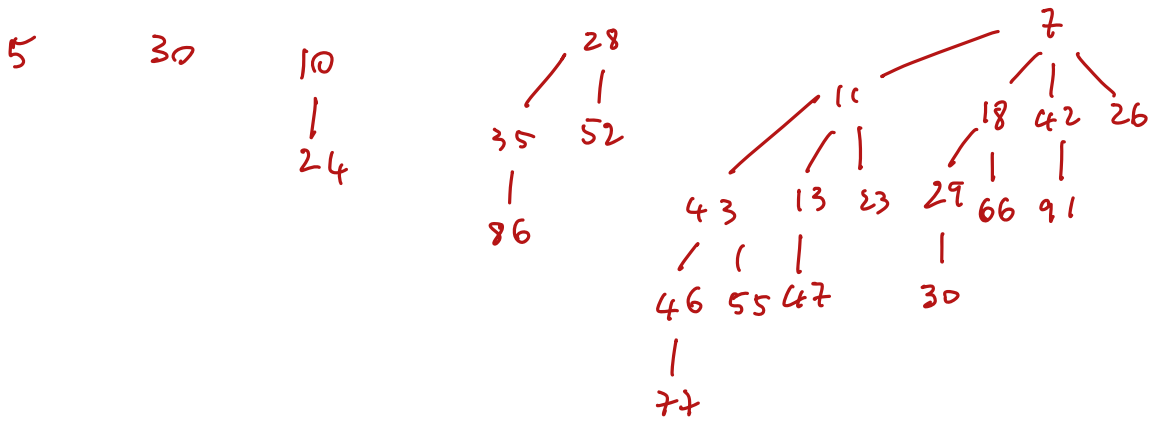
INSERT (30)

30

10
|
24

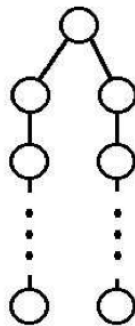


INSERT (5)



ESERCIZIO 1

Trovare una sequenza di operazioni sugli heap di Fibonacci che a partire da una famiglia vuota di heap costruisca un heap formato da un solo albero avente la seguente forma

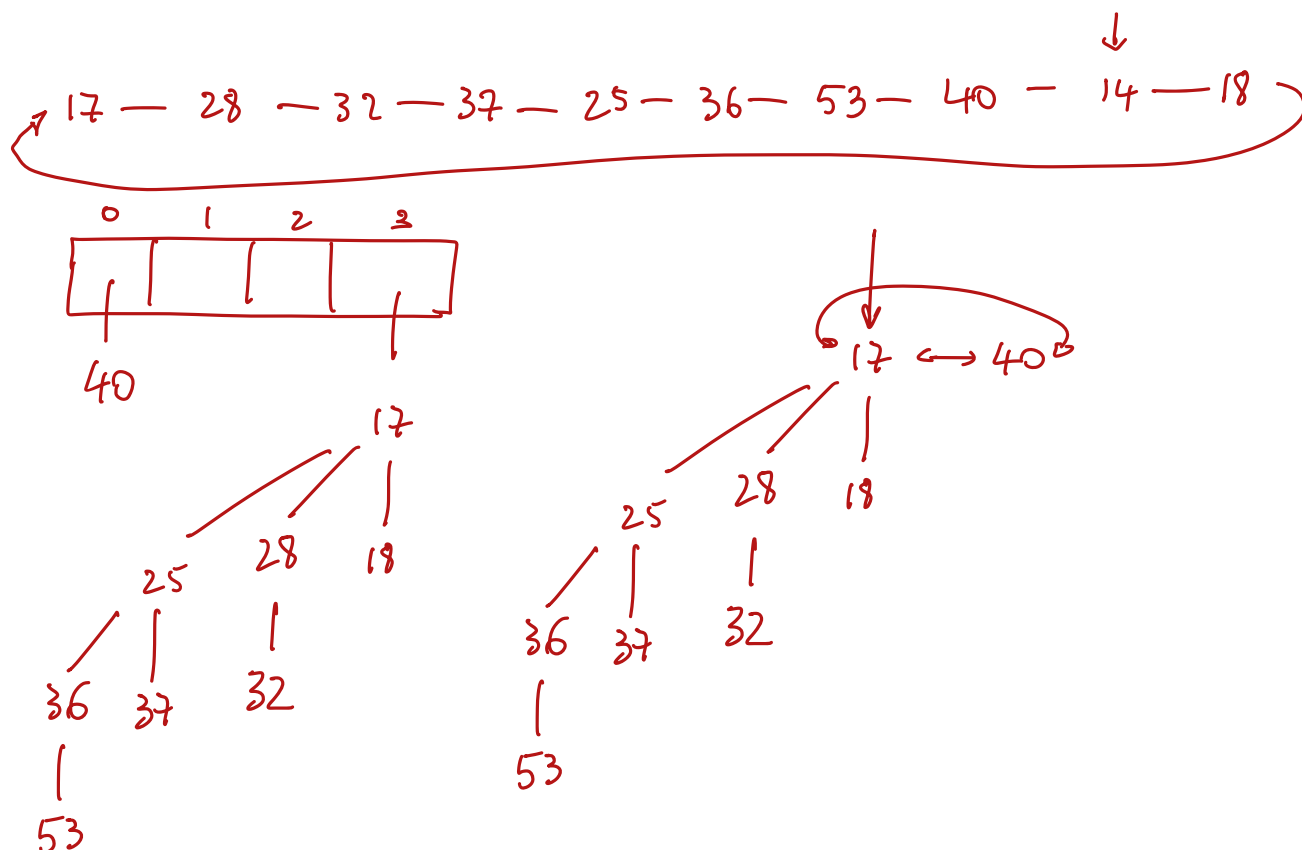


oppure stabilire che una siffatta sequenza di operazioni non esiste.

ESERCIZIO 2 (Heap di Fibonacci)

Si eseguano, nell'ordine dato, le seguenti operazioni su un heap di Fibonacci inizialmente vuoto:

- INSERT 18, 14, 17, 28, 32, 37, 25, 36, 53, 40
- EXTRACT-MIN
- DECREASE-KEY(40,30)
- DELETE(36)
- EXTRACT-MIN



ESERCIZIO 2

Dopo aver descritto gli HEAP DI FIBONACCI, si risponda alle seguenti domande:

- Qual è un limite superiore sul grado massimo di un nodo in un heap di Fibonacci con n nodi?
- Qual è un limite superiore sul grado massimo di un nodo in un heap di Fibonacci con n nodi, costruito a partire da una famiglia vuota di heap senza *mai* usare le operazioni di DECREASE-KEY e di DELETE?
- Stabilire se è corretto o no che l'altezza massima di un albero in un heap di Fibonacci con n nodi sia $\mathcal{O}(\log n)$.

ESERCIZIO 1

Si eseguano nell'ordine dato le seguenti operazioni su un heap di Fibonacci inizialmente vuoto:

- INSERT 27, 17, 19, 20, 24, 12, 11, 10, 14, 18
- EXTRACT-MIN
- DECREASE-KEY(19,7)¹
- DELETE(17)
- DECREASE-KEY(24,5)¹
- EXTRACT-MIN

Dopo ciascuna operazione, si disegni l'heap di Fibonacci risultante, tenendo conto che nella lista delle radici i nuovi elementi vanno inseriti a ~~destra~~ ^{sinistra} del minimo corrente e che il consolidamento della lista delle radici ha sempre inizio nel nodo a destra del minimo appena cancellato.

ESERCIZIO 4

Per ogni $k \geq 5$, trovare una sequenza di operazioni sugli heap di Fibonacci che, a partire da una famiglia vuota di heap, ne costruisca uno formato da un *solo* albero avente la forma riportata nella figura a lato, oppure stabilire che una siffatta sequenza non esiste.

