# In questa lezione

**Analisi della complessità di  una sequenza di operazioni di estrazione e inserimento in una tabella memorizzata su un array, usando l'analisi ammortizzata.**

# Analisi ammortizzata (Tarjan in 1985)

L’analisi ammortizzata comporta il calcolo della media tra i tempi di esecuzione nel caso peggiore di una sequenza di operazioni rispetto al loro numero.

Si applica quando si è interessati al costo di una sequenza di operazioni.

Si può definire quindi il costo ammortizzato di una sequenza di operazioni su un input di dimensione n, $T_a(n)$, come segue:

$$T_a(n) = T(n,k)/k$$

dove T(n,k) è la somma dei tempi nel caso peggiore di k operazioni su un input di dimensione n

# Analisi ammortizzata

**Quindi se molte delle operazioni della sequenza hanno, nel caso peggiore, un costo basso e solo alcune sono molto costose, l'analisi ammortizzata ci dà un risultato di analisi più realistico di quello che otterremmo sommando i costi.**

# Tabella dinamica

**Una tabella dinamica è una struttura dati che supporta le operazioni di inserimento e cancellazione.**

**In alcune applicazioni non si sa  in anticipo quanti elementi dovrà contenere una tabella. Una volta stabilita una sua dimensione ci si potrebbe trovare nella necessità di doverla espandere. Questo comporta la necessità di copiare i dati nella nuova tabella più grande con un costo lineare nel numero degli elementi della tabella.**

**Analogamente se dalla tabella vengono estratti molti elementi sarebbe conveniente ridurne la dimensione.**

**Questo si può fare con costo ammortizzato costante, purché le operazioni di semplice inserimento e semplice cancellazione dalla tabella siano di costo costante.**

**Una pila, una coda o una tabella hash possono essere viste come tabelle dinamiche.**

# Fattore di carico

**Si definisce fattore di carico $\alpha$ di una tabella il rapporto tra il numero degli elementi memorizzati nella tabella e la sua dimensione.**

**Se la tabella ha dimensione zero il suo fattore di carico è 1 per definizione.**

**Per esempio se una pila ha dimensione n e ci sono m elementi allora il fattore di carico è $\alpha = m/n$**

**Esempio: $\alpha = 10/14$**

**S=**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 56 | 75 | 60 | 65 | 30 | 50 | 18 | 40 | 25 | 15 | 6 | 4 | 0 | 7 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

**S.length= 14**

**S.top = 9**

# Espansione 1

Assumiamo che la tabella sia memorizzata in un array.

Come si può ridimensionare l'array in seguito a cancellazioni e inserimenti in modo da avere un costo ammortizzato ragionevole?

Consideriamo prima il caso dell'espansione della tabella

Analizziamo il caso più semplice:

1. per ogni inserimento che "non ha spazio", cioè che porterebbe il fattore di carico a superare 1, aumentiamo la dimensione della tabella di 1.

2. In tal caso se partiamo da una tabella vuota e facciamo n inserimenti abbiamo una complessità di tempo totale pari a

$$1+2+3+4+\ldots+(n-1) = n(n-1)/2 = \Theta(n^2)$$

   quindi un costo ammortizzato $\Theta(n)$.

Non va bene!

# Espansione 2

**Per ogni inserimento che "non ha spazio", cioè che porterebbe il fattore di carico a superare 1, si raddoppia la dimensione della tabella**

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **inserimento** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **...** |
| **costo totale operazione** | **1** | **2** | **3** | **1** | **5** | **1** | **1** | **1** | **9** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **17** | **1** | **1** | **1** | **...** |
| **costo espansione** | | **1** | **2** | | **4** | | | | **8** | | | | | | | | **16** | | | | |

**il costo totale è il costo dell'inserimento, pari a 1 , più il costo dell'espansione, quando necessaria.**

# Analisi aggregata

Partendo da una tabella vuota e facendo n inserimenti, con $2^k < n \leq 2^{k+1}$, si ha un numero di raddoppi pari a

$$1 + 2 + 2^2 + 2^3 + \ldots + 2^k, \text{ dove } 2^k < n \leq 2^{k+1},$$

e quindi un totale in $\Theta(n)$, per un costo ammortizzato $\Theta(1)$!

Si può essere più precisi, considerando il costo totale sommando i costi unitari degli inserimenti e quello dei raddoppi ottenendo

$$1 + 2 + 2^2 + 2^3 + \ldots + 2^k + n = 2^{k+1} - 1 + n \leq 2n - 1 + n < 3n,$$

e quindi un costo ammortizzato < 3.

Il fattore di carico è limitato inferiormente da 1/2,

$$\alpha = m/n \geq 1/2,$$

perché $n/2 \leq m \leq n$ implica $1/2 \leq m/n$

# Contrazioni

**Consideriamo ora anche il caso in cui si voglia diminuire la dimensione di una tabella ogni volta che il numero degli elementi che contiene è molto al di sotto della sua dimensione.**

**Quello che si vuole ottenere, anche in questo caso, è una tabella in cui inserimenti e cancellazioni siano tali da mantenere**

- **il fattore di carico limitato inferiormente da una costante**
- **il costo ammortizzato limitato superiormente da una costante**

**Per gli inserimenti manteniamo la strategia vista di raddoppiare la dimensione della tabella ogni volta che il fattore di carico potrebbe diventare maggiore di 1**

# Contrazioni 1

**Prima soluzione:**

**Per ogni cancellazione che porterebbe il fattore di carico a 1/2, si dimezza la dimensione della tabella**

**Si supponga ora di eseguire, a partire da una tabella vuota, $n = 2^k$ operazioni, di cui le prime n/2 sono inserimenti.**

**Alla fine di questa sequenza la tabella contiene n/2 elementi e la sua dimensione è n/2. Si esegua ora la seguente sequenza**

**Ins, Del, Del, Ins, Ins, Del, Del, Ins, Ins, ... ,**

**Il primo inserimento causa un'espansione della tabella a una di n elementi, ma le due cancellazioni successive comportano una contrazione di nuovo a n/2.**
**La cosa si ripete con i successivi inserimenti!**
**La sequenza di n/2 operazioni comporta $\Theta(n)$ espansioni e contrazioni per un costo totale di $\Theta(n^2)$, e uno ammortizzato $\Theta(n)$.**

**Non si è raggiunto il secondo obiettivo di mantenere il costo ammortizzato limitato superiormente da una costante.**

# Contrazioni 2

**Seconda soluzione:**

- **per ogni cancellazione che porterebbe il fattore di carico a meno di 1/4, si dimezza la dimensione della tabella**

**Dopo un'espansione il fattore di carico è 1/2, quindi metà degli elementi della tabella devono essere cancellati prima che si debba di nuovo contrarre la tabella, cosa che avviene quando il fattore di carico rischia di diventare più piccolo di 1/4.**

**D'altro canto dopo una contrazione il fattore di carico è 1/2, perché se si ha dimensione n, ci sono n/4 elementi e si riduce la dimensione di (1/2)n allora il fattore di carico è (n/4)/(n/2) = 1/2.**

**Quindi ancora il numero degli elementi deve raddoppiare prima che sia necessaria una nuova espansione**

# Contrazione 2

**Per ogni cancellazione che porterebbe il fattore di carico a meno di 1/4, si dimezza la dimensione della tabella**

| elementi inseriti | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| costo cancellazione | | | | 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| costo contrazione | | | | 4 | | | | | | | | | | | | |

**il costo totale è il costo di una cancellazione, pari a 1 , più il costo della contrazione, quando necessaria.**

# Conclusione

**Si può dimostrare che il costo ammortizzato di ogni operazione di cancellazione o inserimento è superiormente limitato da una costante, possiamo concludere che la sequenza di n operazioni in una tabella dinamica è O(n).**

# Applicazione alla coda

In caso di coda piena, cioè quando Q.rear +1 = Q.front

| 56 | 75 | 60 | 65 | 30 | 50 | 18 | 40 | 25 | 15 | 6 | 4 | 0 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

Q.size= 14

Q.rear = 9 Q.front = 10

Si modifica l'inserimento in coda in modo da raddoppiare la dimensione della coda, prima di inserire il nuovo elemento. Si pone Q.size a 2Q.size, e gli elementi vengono ricopiati all'inizio del nuovo array, per cui front = 0 e rear = Q.size

# Applicazione alla coda

Per ridimensionare con le contrazioni, quando il fattore di carico rischia di scendere al di sotto di 1/4, occorre calcolare

dim = (Q.rear - Q.front + Q.size) % Q.size, il numero degli elementi, e verificare se dim ≤ Q.size/4

Esempio: dim = 3 e 3 ≤ 14/4 = 3,5

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 56 | 75 | 60 | 65 | 30 | 50 | 18 | 40 | 25 | 15 | 6 | 4 | 55 | 66 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

Q.size= 14

Q.rear = 1

Q.front = 12

Si potrebbe anche aggiungere un controllo per non dimezzare quando si scende al di sotto di una soglia, per esempio 1/4 della dimensione iniziale della coda.