Tabelle Hash

Soluzione standard per il problema del dizionario

Tabelle Hash

Soluzione standard per il problema del dizionario

Mantenere un insieme $S \subseteq U$ e fornire le seguenti operazioni

Insert(x): inserisce x in S

Delete(x): rimuove x da S

Search(x): determina se x è in S

Tabelle Hash

Soluzione standard per il problema del dizionario

Mantenere un insieme $S \subseteq U$ e fornire le seguenti operazioni

Insert(x): inserisce x in S

Delete(x): rimuove x da S

Search(x): determina se x è in S

Vedremo Tabelle hash con gestione dei conflitti con liste di adiacenza

Tabelle Hash

Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S$\rightarrow$|T|

h(5)=1

h(10)=h(22)=h(33)=3

h(15)=h(20)=7

h(25)=9

Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S$\rightarrow$|T|

h(5)=1

h(10)=h(22)=h(33)=3

h(15)=h(20)=7

h(25)=9

T

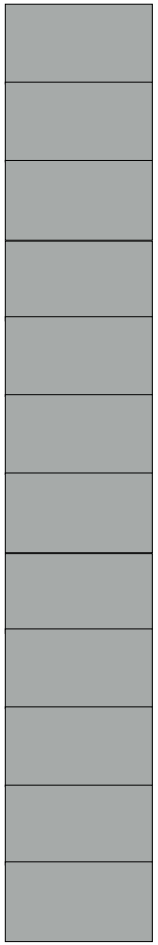


Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S$\rightarrow$|T|

h(5)=1
h(10)=h(22)=h(33)=3
h(15)=h(20)=7
h(25)=9

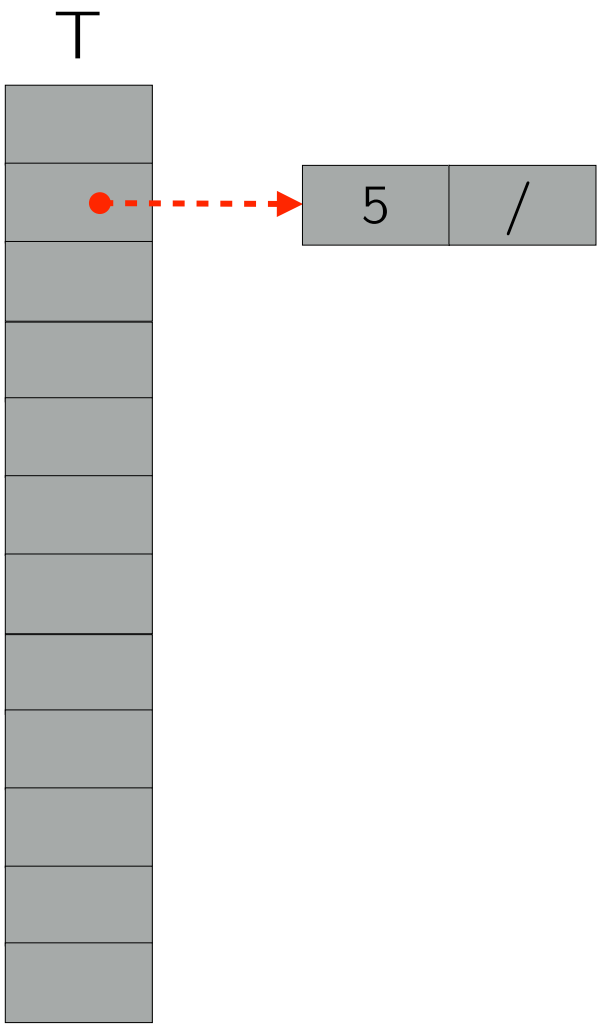


Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S$\rightarrow$|T|

h(5)=1
h(10)=h(22)=h(33)=3
h(15)=h(20)=7
h(25)=9

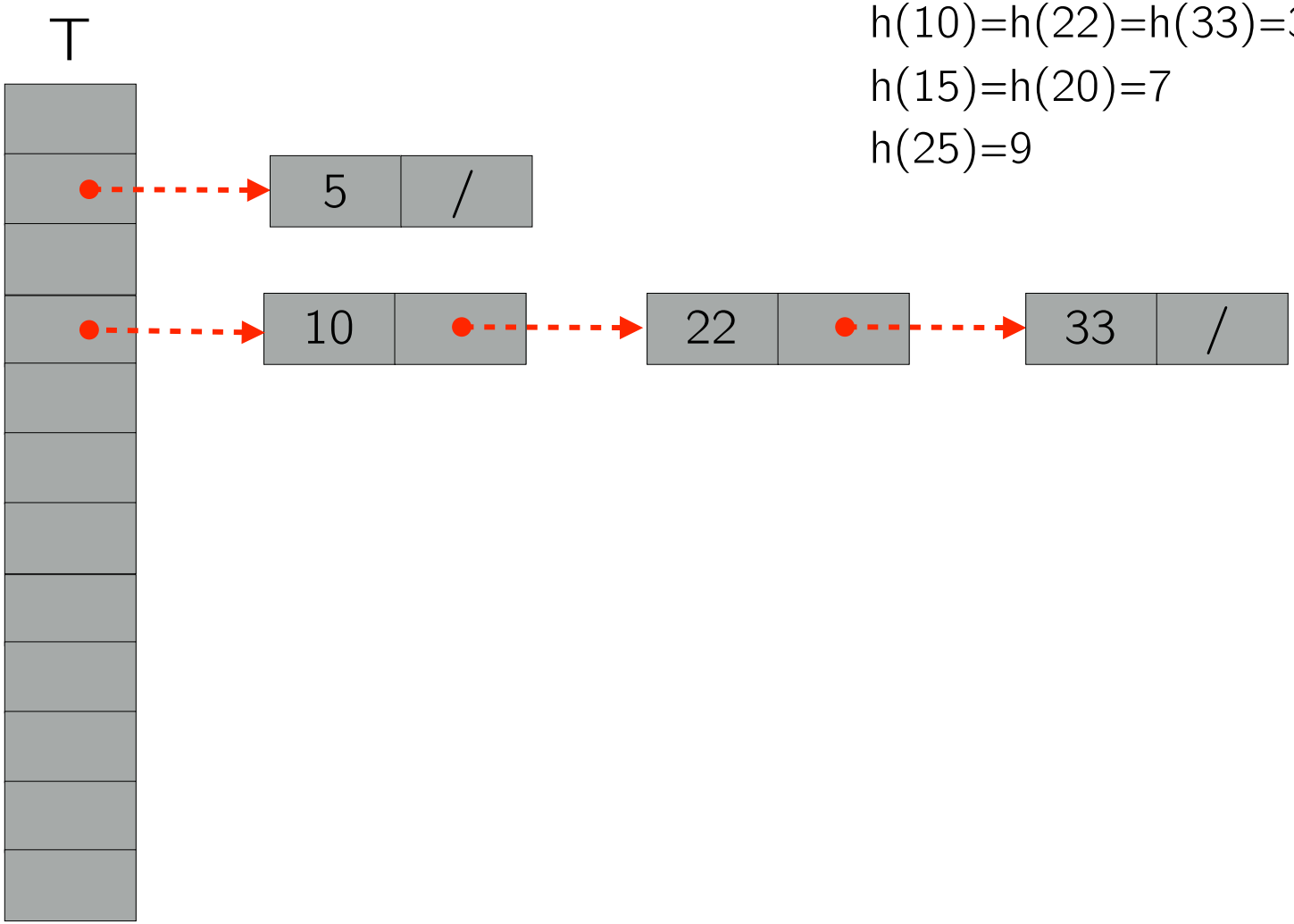


Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S$\rightarrow$|T|

h(5)=1
h(10)=h(22)=h(33)=3
h(15)=h(20)=7
h(25)=9

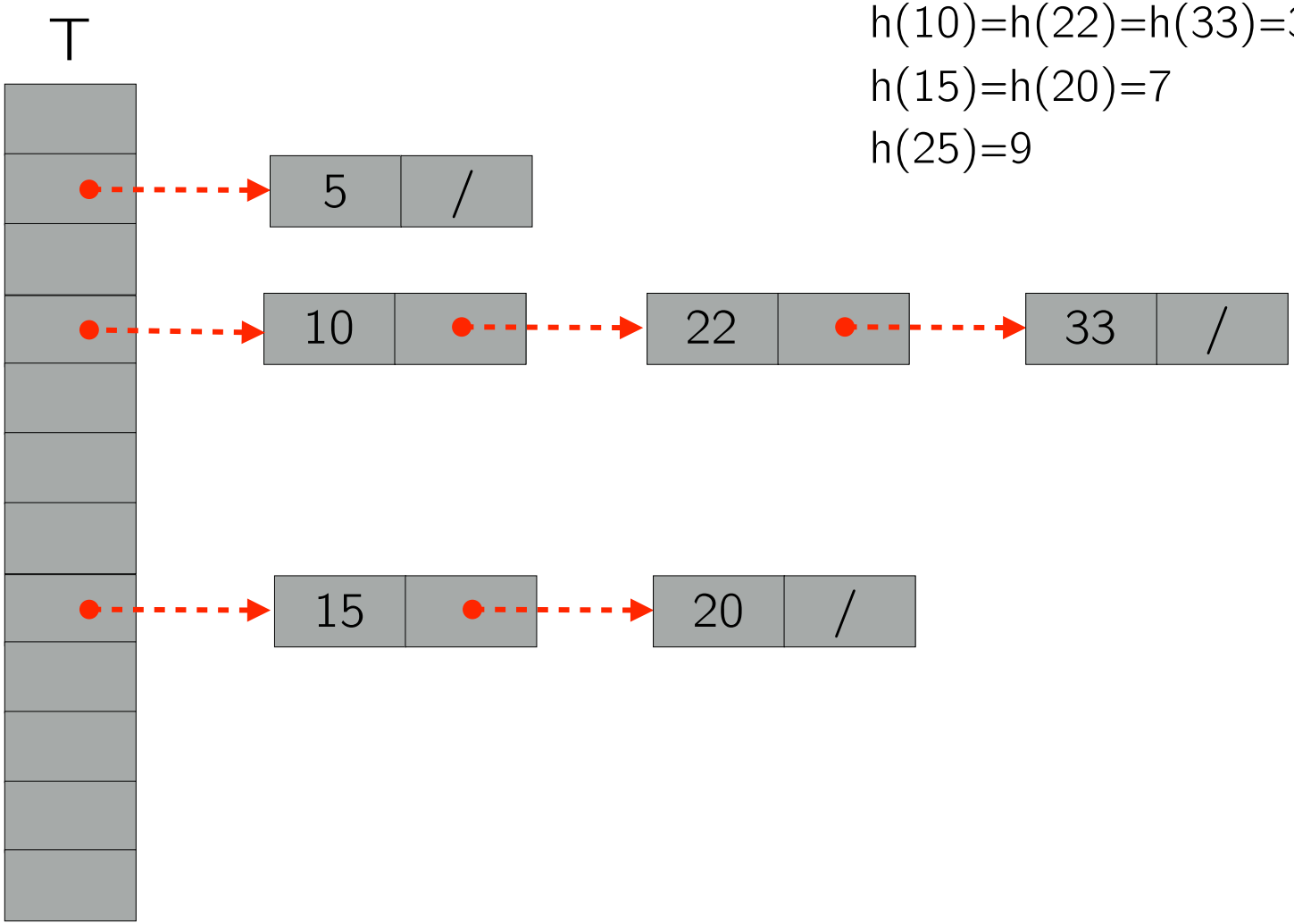


Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S$\rightarrow$|T|

h(5)=1
h(10)=h(22)=h(33)=3
h(15)=h(20)=7
h(25)=9

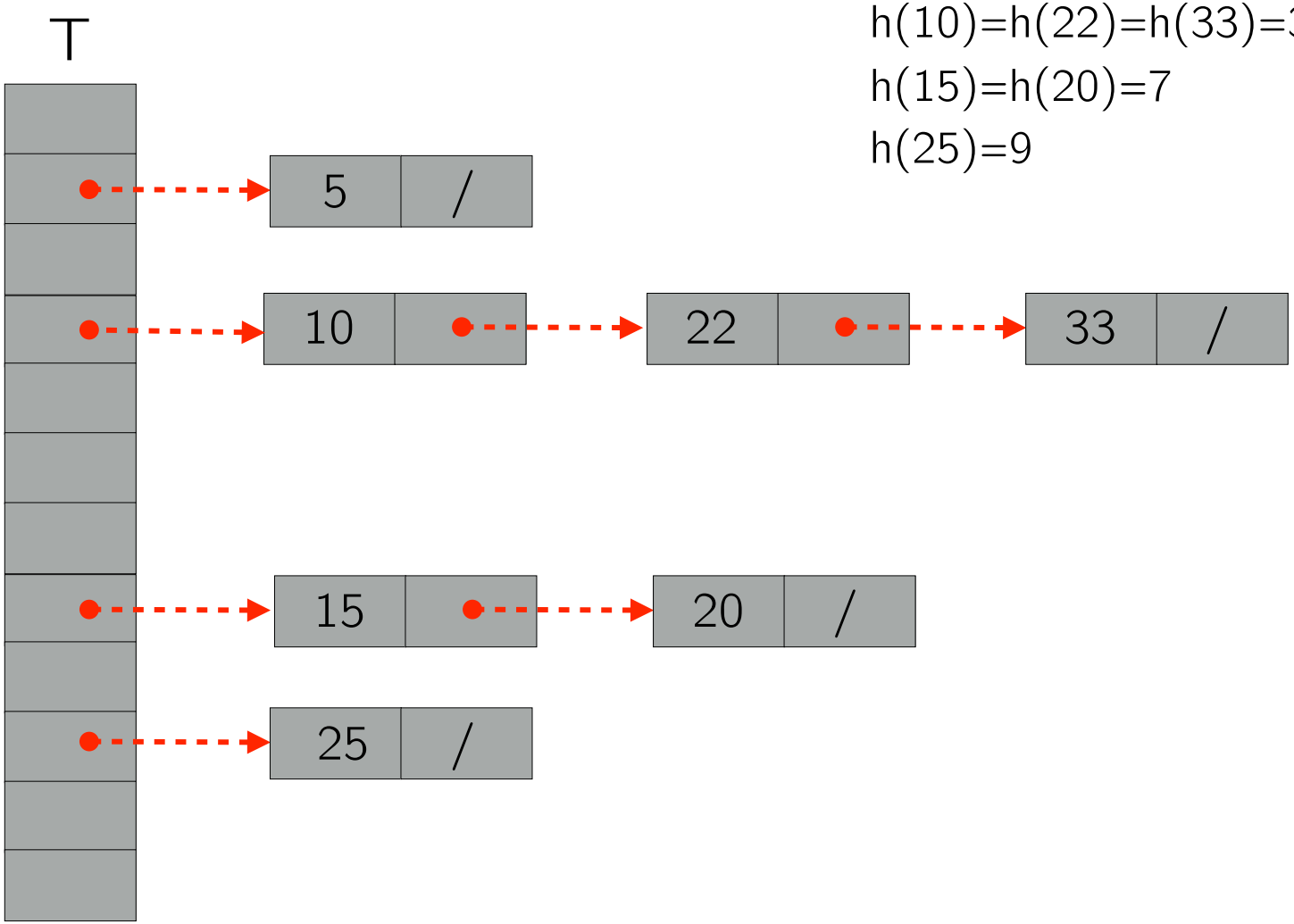
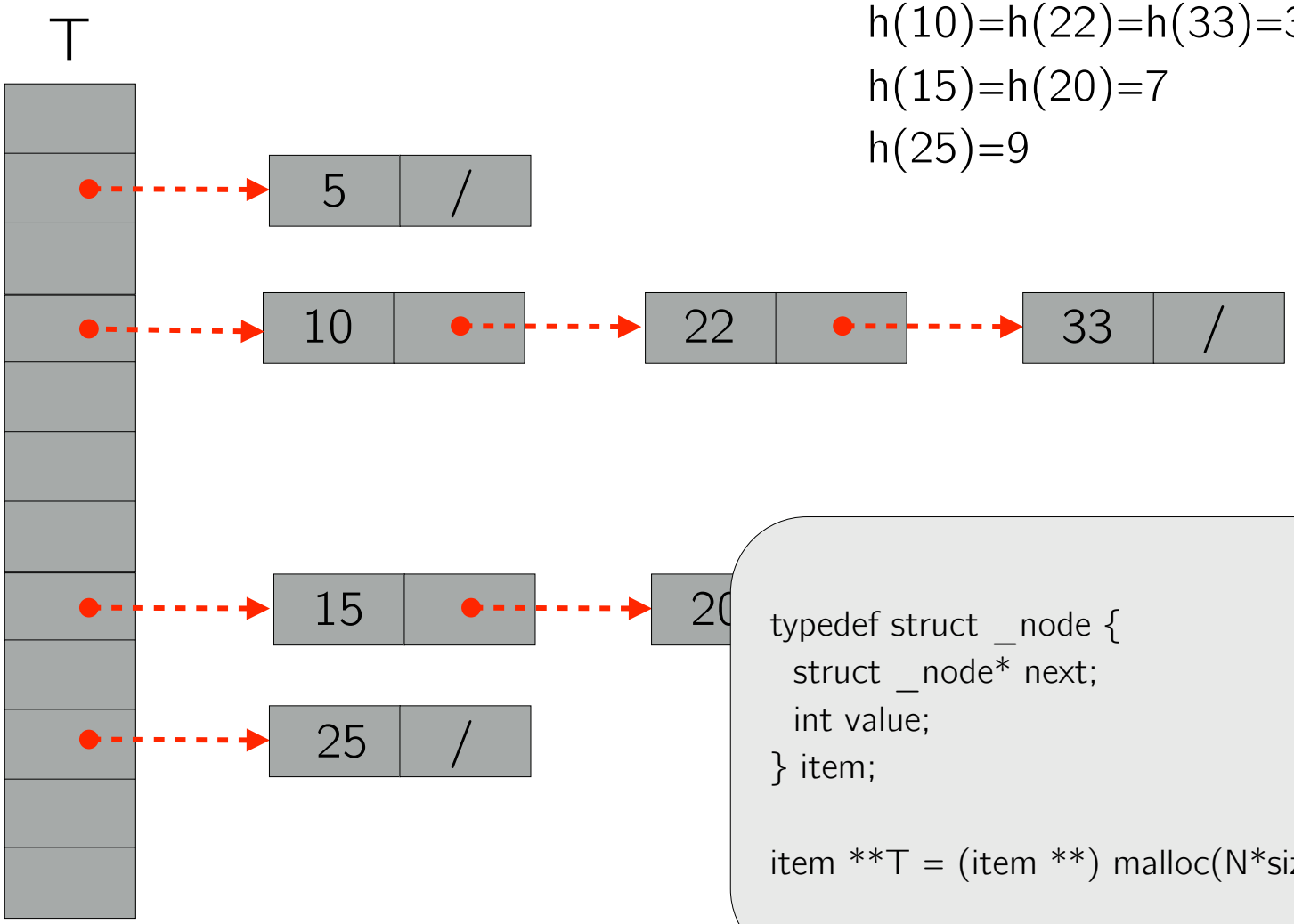


Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S$\rightarrow$|T|

h(5)=1
h(10)=h(22)=h(33)=3
h(15)=h(20)=7
h(25)=9




```
typedef struct _node {
  struct _node* next;
  int value;
} item;

item **T = (item **) malloc(N*sizeof(item *));
```

Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S→|T|

h(5)=1

(33)=3

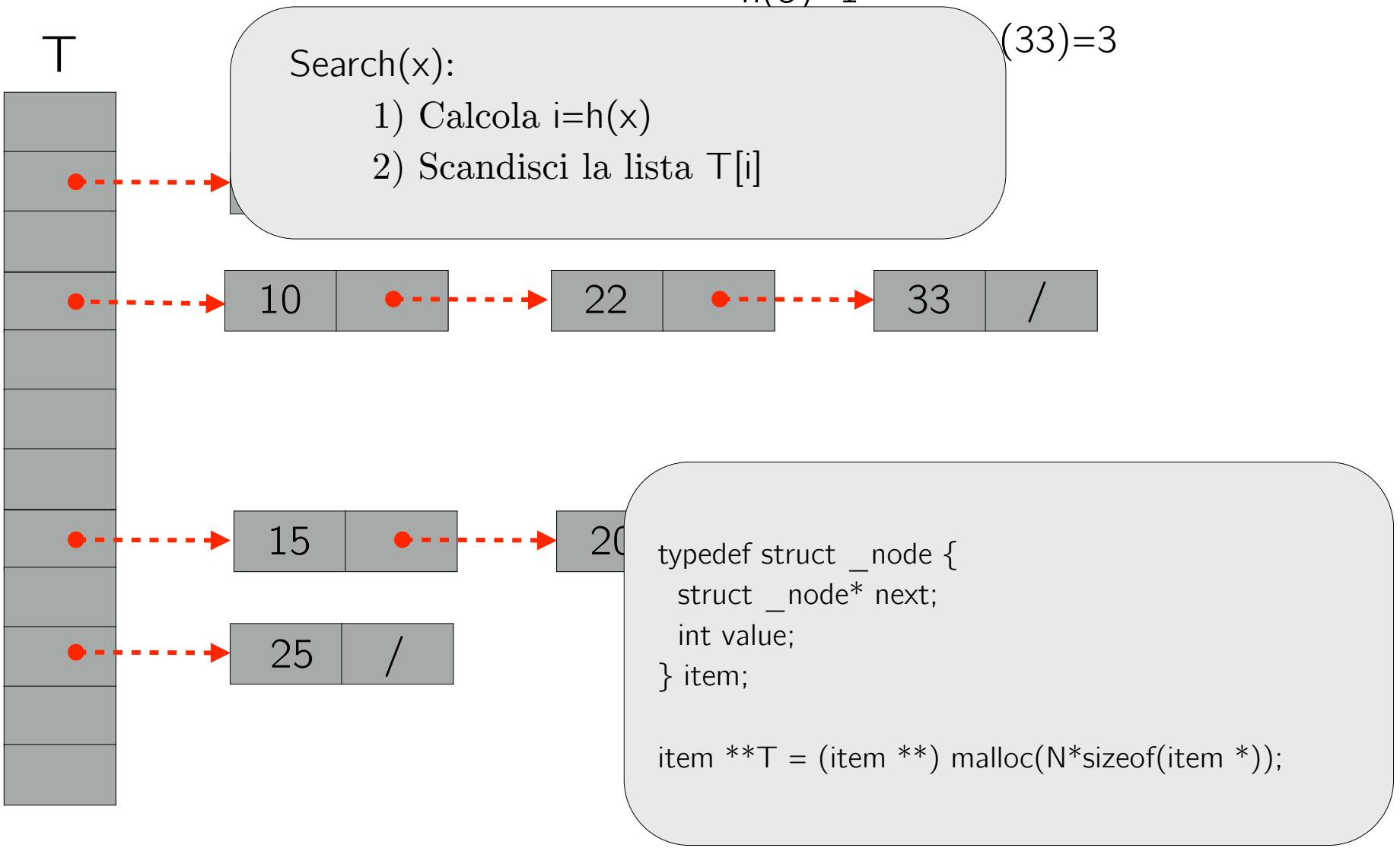


Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S→|T|

h(5)=1

(33)=3

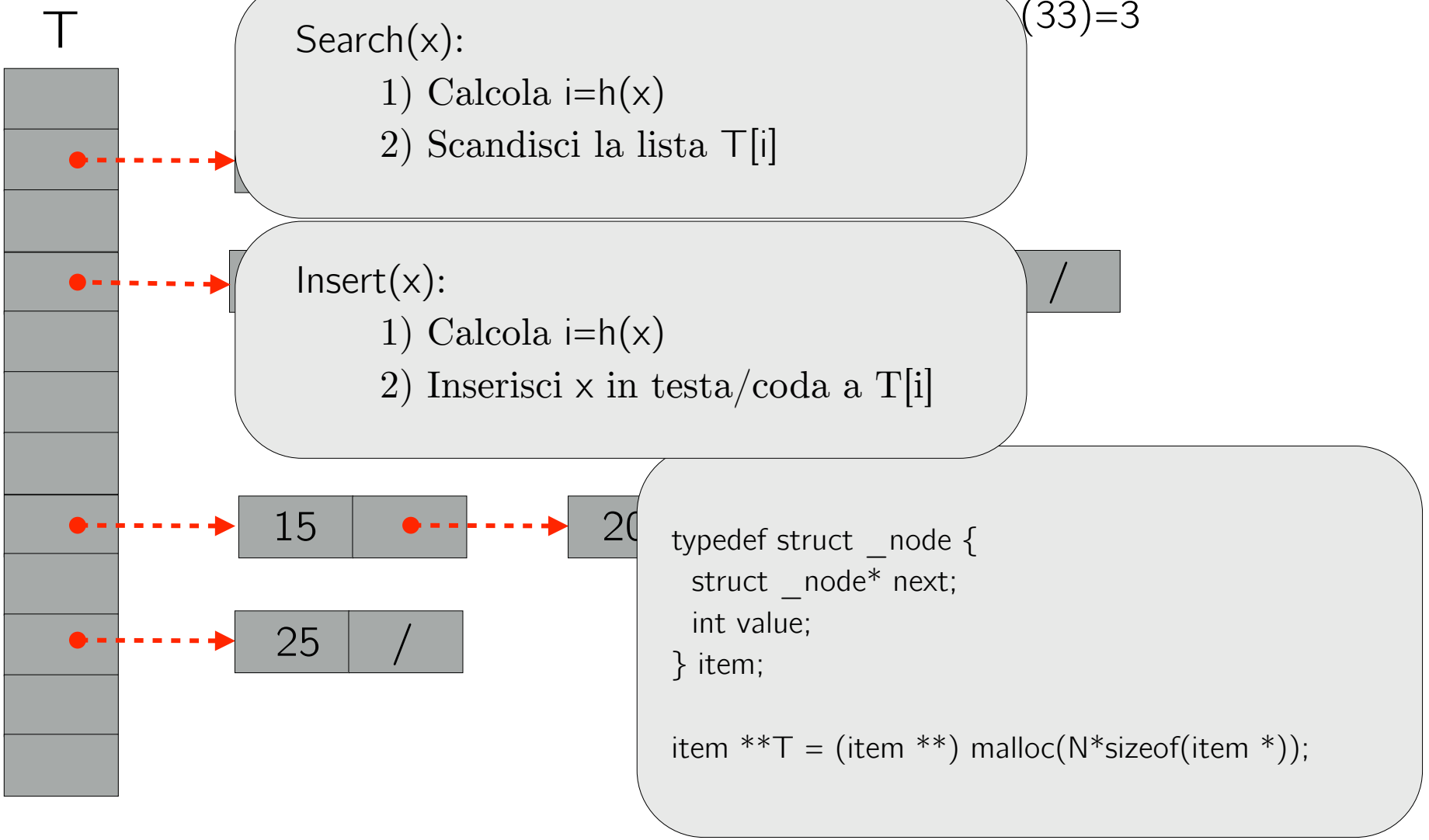


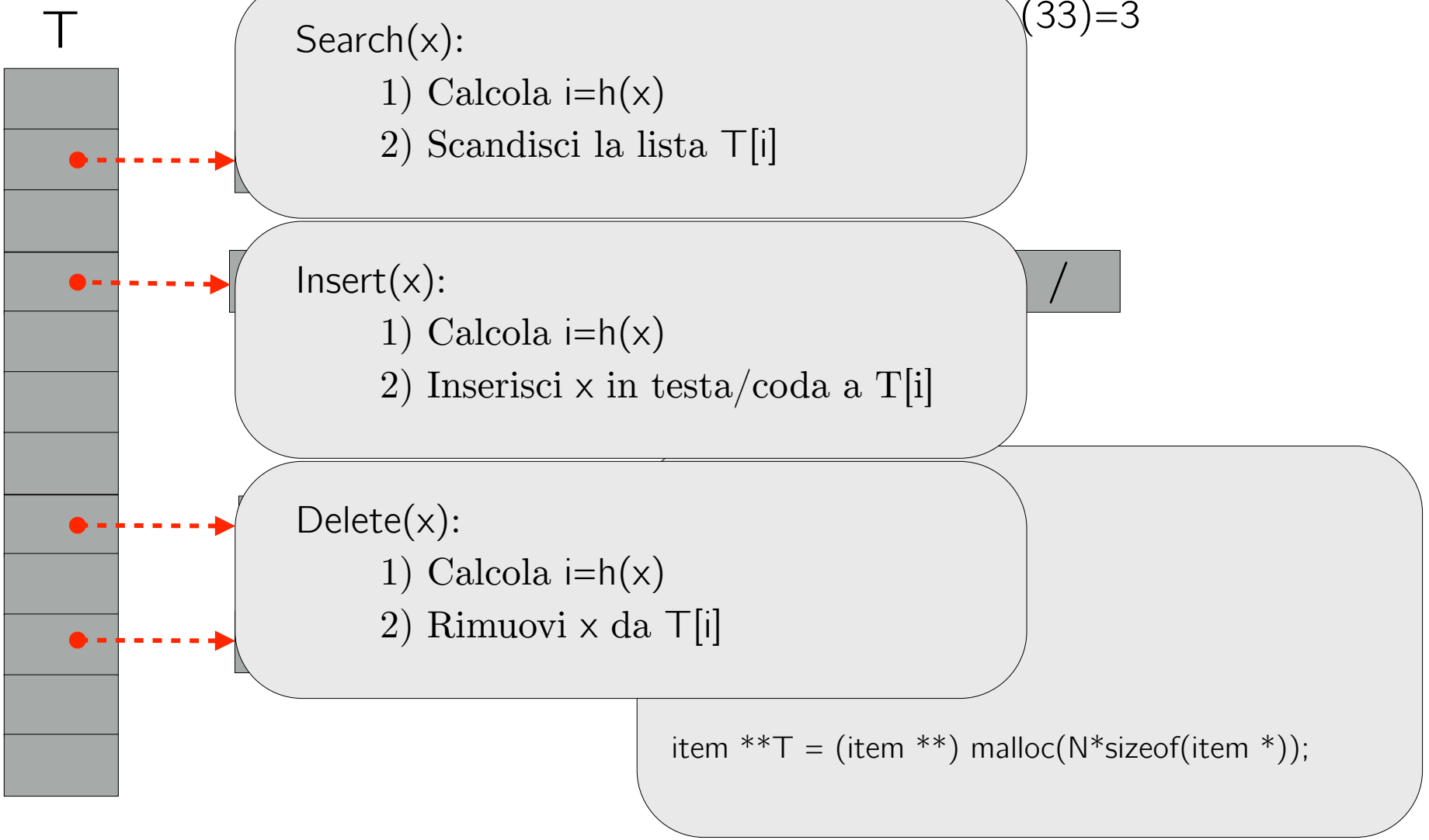
Tabelle Hash

S={5, 10, 15, 20, 22, 25, 33}

h: S→|T|

h(5)=1

(33)=3



Esercizio 1

Tabelle Hash: inserimento

Scrivere un programma che legga da tastiera una sequenza di n interi **distinti** e li inserisca in una tabella hash di dimensione $2n$ posizioni utilizzando liste monodirezionali per risolvere eventuali conflitti.

Utilizzare la funzione hash $h(x) = ((ax + b) \ \% \ p) \ \% \ 2n$ dove p è il numero primo 999149 e a e b sono interi positivi minori di 10.000 scelti casualmente.

Una volta inseriti tutti gli interi, il programma deve stampare la lunghezza massima delle liste e il numero totale di conflitti.

Prima di scrivere il programma chiedersi perché la tabella ha dimensione $2n$ e non n.

Esercizio 2

Tabelle Hash: inserimento con rimozione dei duplicati

Scrivere un programma che legga da tastiera una sequenza di n interi **NON distinti** e li inserisca senza duplicati in una tabella hash di dimensione $2n$ posizioni utilizzando liste monodirezionali per risolvere eventuali conflitti.

Utilizzare la funzione hash $h(x) = ((ax + b) \ \% \ p) \ \% \ 2n$ dove p è il numero primo 999149 e a e b sono interi positivi minori di 10.000 scelti casualmente.

Una volta inseriti tutti gli interi, il programma deve stampare il numero totale di conflitti, la lunghezza massima delle liste e il numero di elementi distinti.