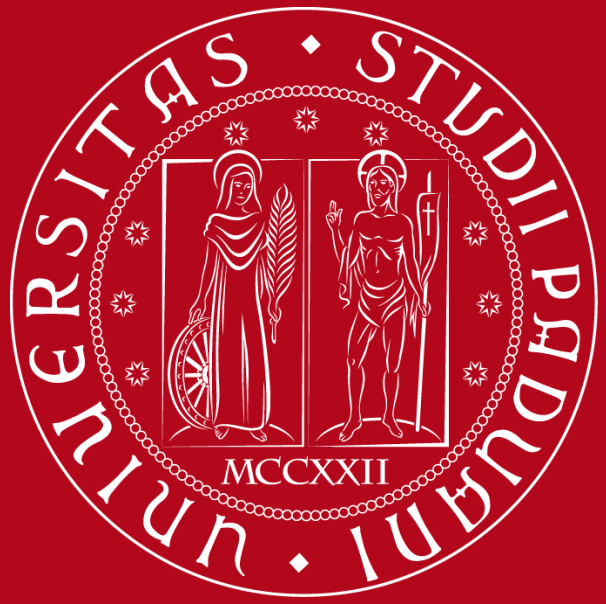


UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Associative container

Stefano Ghidoni

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Agenda

- Associative container
  - Map
  - Set
- Gestione di una coppia: pair

Associative container – lista

Contenitore	Significato
map	Contenitore ordinato di coppie (chiave, valore)
set	Contenitore ordinato di chiavi
unordered_map	Contenitore non ordinato di coppie (chiave, valore)
unordered_set	Contenitore non ordinato di chiavi
multimap	Una map in cui una chiave può ricorrere più di una volta
multiset	Un set in cui una chiave può ricorrere più di una volta
unordered_multimap	unordere_map in cui una chiave può ricorrere più di una volta
unordered_multiset	unordered_set in cui una chiave può ricorrere più di una volta

map

map

- Gestisce coppie chiave-valore
- Contenitore ordinato
- Ottimizzato per ricerche dalla chiave
  - È facile e veloce verificare se una chiave è presente e recuperare il relativo valore
- Rappresentato con un albero bilanciato
- Header: <map>

map – esempio

- Esempio: creare un istogramma delle parole in un testo

```
int main()
{
    map<string, int> words;

    for(string s; cin >> s; )
        ++words[s];

    for(const auto& p : words)
        cout << p.first << ": " << p.second << '\n';

    return 0;
}
```

Elemento cruciale

Ciclo sugli elementi di words

Indicizzare map

```
for(string s; cin >> s; )
    ++words[s];
```

- Indicizzazione con una string
- A partire da string, la map fornisce il relativo int
- Se la chiave non esiste nella map, è creato un elemento con quella chiave
  - Il corrispondente int è inizializzato a 0 – costruttore di default di int

pair

- Gli elementi della map sono coppie chiave-valore
  - Gestite come pair<string, int>
  - Elementi di una pair: first e second

```
cout << p.first << ": " << p.second << '\n';
```

- Per creare una pair: make_pair()

map vs vector

- map ordina i dati rispetto alla chiave
- map offre una ricerca molto più veloce a partire dalla chiave
- Associazione esplicita di una chiave a un valore

set

set

- Un set è una map senza i valori
- Viene meno la ricerca del valore a partire dalla chiave
  - Non è disponibile operator[]
  - Non è disponibile push_back() – è il set che decide dove inserire l'elemento
  - Gestito tramite le operazioni tipiche delle liste
    - insert() ed erase()
- Header: <set>

set

- set è un contenitore ordinato
- Deve essere definita una funzione d'ordine per ciascun tipo contenuto
  - Inclusi UDT, ovviamente

set – esempio

- Es: se vogliamo costruire un set di elementi Fruit:

```
struct Fruit{
    string name;
    int count;
    double unit_price;
    Date last_sale_date;
    // ...
};
```

set – esempio

- È necessario definire una funzione d'ordine:

```
struct Fruit_order {
    bool operator()(const Fruit& a, const Fruit& b) const
    {
        return a.name < b.name;
    }
};
```

- È ora possibile creare:

```
set<Fruit, Fruit_order> inventory;
// Nota: Fruit_order serve per confrontare Fruit
```

set – esempio

- È possibile popolare il set così:

```
inventory.insert(Fruit{"Quince", 5});
inventory.insert(Fruit{"Apple", 200, 0.37});
```

- E scorrerlo / stamparlo così:

```
for(auto p = inventory.begin(); p != inventory.end(); ++p)
    cout << *p << '\n';
```

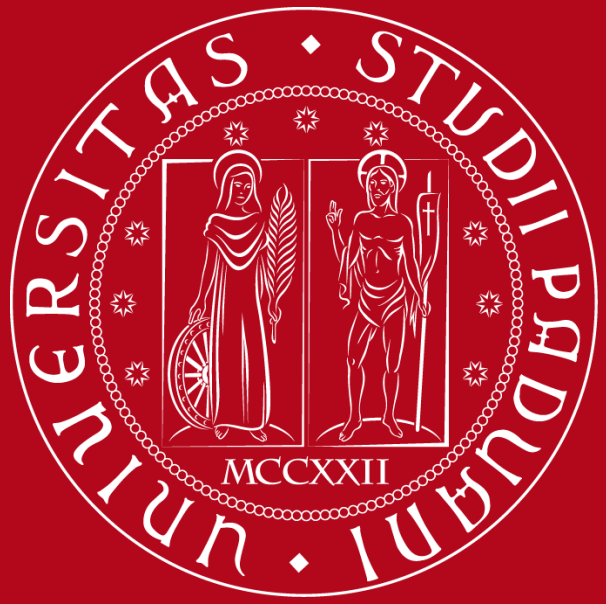
  - map non contiene pair, si accede al dato direttamente con *

Spunti di approfondimento

- Unordered map
  - Implementano le hash table
- Stream iterator

Recap

- Uso di map per la gestione di coppie chiave-valore
- Uso di set per la gestione ordinata di elementi
- Funzioni d'ordine
- Inserimento di oggetti in container ordinati



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Associative container

Stefano Ghidoni

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE