

1. Hàm `push_back()` và `pop_back()`

Hàm `push_back()` sẽ thêm 1 ký tự vào cuối string và `pop_back()` sẽ xóa đi ký tự cuối cùng trong string.

Lưu ý là hàm này sẽ thêm 1 ký tự chứ không phải là thêm 1 chuỗi ký tự, nếu bạn cố tính `push_back` 1 chuỗi ký tự sẽ gây lỗi.

Độ phức tạp của 2 hàm này đều là $O(1)$

Ví dụ :

```
#include <iostream>
#include <string>

using namespace std;

int main(){
    string s = "28tech";
    s.push_back('@');
    cout << s << endl;
    s.push_back('#');
    cout << s << endl;
    s.pop_back();
    cout << s << endl;
}
```

Output :

```
28tech@
28tech@#
28tech@
```

2. Hàm `append()`

Hàm `append()` có chức năng nối chuỗi hoặc nối ký tự vào cuối của string. Bạn có thể lựa chọn nối toàn bộ chuỗi khác vào chuỗi hiện tại hoặc nối 1 vài ký tự liên tiếp của xâu khác vào xâu hiện tại.

Độ phức tạp : $O(N)$

Ví dụ :

```
#include <iostream>
#include <string>

using namespace std;

int main(){
    string s = "28tech";
    //Nối toàn bộ xâu
    s.append("blog");
    cout << s << endl;
    string t = "0123456789";
    //Nối 5 kí tự tính từ chỉ số 1 của xâu t vào s
    s.append(t, 1, 5);
    cout << s << endl;
}
```

Output :

28techblog

28techblog12345

Có một cách khác để nối chuỗi nhanh hơn trong string đó là sử dụng toán tử `+=`, toán tử này khi áp dụng với string sẽ tương đương phép nối xâu.

```
#include <iostream>
#include <string>

using namespace std;

int main(){
    string s = "28tech";
    //Nối toàn bộ xâu
    s += " blog";
    cout << s << endl;
}
```

```

//Nối 1 ký tự
s += '@';
cout << s << endl;
s += " C++";
cout << s << endl;
}

```

Output :

```

28tech blog
28tech blog@
28tech blog@ C++

```

3. Hàm insert()

Hàm **insert()** giúp bạn chèn một xâu khác vào vị trí bất kỳ trong xâu hiện tại thông qua chỉ số chèn.

Cú pháp : s.insert(index, string t);

Trong đó xâu t được chèn vào chỉ số index của xâu s, hàm này có tương đối nhiều cách dùng tuy nhiên chèn vào vị trí xác định trong xâu là phổ biến nhất.

Độ phức tạp : O(N)

Ví dụ :

```

#include <iostream>
#include <string>

using namespace std;

int main(){
    string s = "28tech";
    //Chen vao chi so 2 cua xau s
    s.insert(2, "####");
    cout << s << endl;
    //Chen vao dau xau s
}

```

```
s.insert(0, "blog ");  
cout << s << endl;  
}
```

Output :

```
28####tech  
blog 28####tech
```

4. Hàm erase

Hàm **erase()** giúp bạn có thể xóa đi 1 vài ký tự của xâu hiện tại thông qua chỉ số.

Cú pháp :

- s.erase(pos) : Xóa tất cả các ký tự trong xâu s từ chỉ số pos tới cuối xâu
- s.erase(pos, n) : Xóa n ký tự trong xâu s tính từ chỉ số pos

Độ phức tạp : O(N)

Ví dụ :

```
#include <iostream>  
#include <string>  
  
using namespace std;  
  
int main(){  
    string s = "28tech.com.vn";  
    //Xoa tu chi so 6 toi het xau  
    s.erase(6);  
    cout << s << endl;  
    string t = "28tech.com.vn";  
    //Xoa 4 ki tu tinh tu chi so 2  
    t.erase(2, 4);  
    cout << t << endl;  
}
```

Output :

28tech
28.com.vn

5. Hàm substr()

Hàm **substr()** giúp bạn có thể tách xâu con của xâu hiện tại, hàm này trả về xâu con tách được.

Cú pháp :

- s.substr(pos) : Trả về xâu con của xâu s tính từ chỉ số pos
- s.substr(pos, n) : Trả về xâu con gồm n ký tự tính từ chỉ số pos của xâu s

Độ phức tạp : O(N)

Ví dụ :

```
#include <iostream>
#include <string>

using namespace std;

int main(){
    string s = "28tech.com.vn";
    //Xau con tu chi so 2 toi het xau
    string sub1 = s.substr(2);
    cout << sub1 << endl;
    string t = "28tech.com.vn";
    //Xau con tu chi so 0 co do dai 6
    string sub2 = t.substr(0, 6);
    cout << sub2 << endl;
}
```

Output :

tech.com.vn
28tech

6. Hàm find()

Hàm **find()** giúp bạn tìm kiếm sự xuất hiện của xâu con trong xâu hiện tại.

Hàm này sẽ trả về chỉ số đầu tiên của xâu con nếu xâu này tồn tại trong xâu đang tìm kiếm, ngược lại sẽ trả về giá trị là **string::npos**

Độ phức tạp : $O(N)$

Ví dụ :

```
#include <iostream>
#include <string>

using namespace std;

int main(){
    string s = "28tech.com.vn 28tech";
    if(s.find("28tech") == string::npos){
        cout << "NOT FOUND\n";
    }
    else{
        cout << "FOUND\n";
        cout << s.find("28tech");
    }
}
```

Output :

FOUND

0