

SAU BÀI HỌC NÀY EM SẼ:

- Biết được cách đọc dữ liệu từ tệp dữ liệu trong máy tính.
- Áp dụng được thuật toán tìm kiếm trong một vài bài toán cụ thể.



Trong bài học trước, các em đã được học cách triển khai thuật toán tìm kiếm tuần tự và tìm kiếm nhị phân; vậy chúng ta có thể áp dụng các thuật toán tìm kiếm đã học vào thực tế như thế nào? Trong các bài toán thực tế, các dữ liệu thường không được viết trực tiếp vào chương trình mà thường được lưu trong các tệp chứa dữ liệu. Ở bài học này, chúng ta sẽ được học cách đọc dữ liệu từ tệp chứa dữ liệu trong máy tính.



Nhiệm vụ 1. Đọc và ghi tệp trong Python

Hướng dẫn:

Để thực hiện được các bài tập thực hành trong sách, em cần biết một số lệnh và thao tác đọc, ghi dữ liệu từ tệp văn bản.

a) Đối tượng tệp

Python cho phép đọc, ghi dữ liệu với tệp văn bản (text file) rất dễ dàng. Các tệp đều được mở để đọc và ghi dưới dạng văn bản. Đối tượng tệp được tạo ra bằng lệnh `open()` với các cách như sau:

– Lệnh mở tệp để đọc:

```
f = open(<file name>, "r", encoding = "UTF-8")
```

Chú ý, tham số "r" có thể bỏ qua. Tham số encoding dùng khi mở tệp văn bản có mã Unicode, ví dụ tiếng Việt. Nếu không có tham số encoding thì chỉ đọc được văn bản có mã ASCII.

– Lệnh mở tệp để ghi dữ liệu từ đầu:

```
f = open(<file name>, "w", encoding = "UTF-8")
```

Chú ý, tham số "w" là bắt buộc.

– Lệnh mở tệp để ghi tiếp dữ liệu:

```
f = open(<file name>, "a", encoding = "UTF-8")
```

Chú ý, tham số "a" là bắt buộc.

Sau khi mở tệp thì có thể tiến hành đọc (hoặc ghi) dữ liệu. Sau khi thực hiện xong các thao tác đọc, ghi thì cần đóng đối tượng tệp bằng lệnh sau:

```
f.close()
```

b) Các lệnh đọc dữ liệu từ tệp văn bản

Dữ liệu trong các tệp văn bản được lưu dưới dạng các dòng, mỗi dòng là một chuỗi kí tự. Sau khi mở tệp bằng lệnh `open()`, con trỏ đọc mặc định ở đầu của tệp sẵn sàng chờ lệnh đọc.

Bảng 20.1 mô tả và minh họa kết quả của các lệnh đọc dữ liệu từ đối tượng `f` trong Python với tệp `Data.inp`.

Data.inp
Hà 9.6
Bình 8.5
Quang 7.2

Bảng 20.1. Các lệnh đọc dữ liệu từ đối tượng `f` trong Python

Lệnh đọc dữ liệu	Ý nghĩa	Kết quả dữ liệu
<code>f.readline()</code>	Đọc một dòng tiếp theo từ <code>f</code> .	Kết quả là từng dòng của tệp <code>f</code> .
<code>f.readlines()</code>	Đọc toàn bộ dữ liệu tệp, đưa kết quả vào một danh sách (list), mỗi phần tử là một dòng.	['Hà 9.6\n', 'Bình 8.5\n', 'Quang 7.2']
<code>L = list(f)</code>	Đưa toàn bộ nội dung <code>f</code> vào list <code>L</code> , mỗi dòng là một phần tử.	['Hà 9.6\n', 'Bình 8.5\n', 'Quang 7.2']
<code>for line in f:</code>	Duyệt từng dòng của <code>f</code> .	Lệnh này tương đương nhóm lệnh: <code>L = list(f)</code> <code>for line in L:</code>

Ví dụ 1. Với bộ dữ liệu trên, cần đọc và đưa vào hai list là `TenHS` và `DiemHS`. Chương trình sau đọc dữ liệu từ tệp `Data.inp`, đưa dữ liệu vào hai mảng trên và hiển thị kết quả ra màn hình.

```
1 fname = "Data.inp"
2 def NhapDL(fname):
3     f = open(fname, encoding = "UTF-8")
4     TenHS = []
5     DiemHS = []
6     for line in f:
7         L = line.split()
8         TenHS.append(L[0])
9         DiemHS.append(float(L[1]))
10    f.close()
11    return TenHS, DiemHS
12
13 TenHS, DiemHS = NhapDL(fname)
14 for i in range(len(TenHS)):
15     print(TenHS[i], DiemHS[i])
```


c) Ghi dữ liệu ra tệp văn bản

Việc ghi dữ liệu ra tệp văn bản đơn giản nhất là sử dụng lệnh `print()`. Cú pháp lệnh `print()` như sau:

```
print(<v1>, <v2>, ..., file = <đối tượng file>)
```

Chú ý, tham số `file = <đối tượng file>` là bắt buộc khi cần đưa dữ liệu ra tệp. Tất cả các tham số khác của lệnh `print()` vẫn có giá trị khi đưa dữ liệu ra tệp.

Ví dụ 2. Giả sử có hai mảng `TenHS` và `DiemHS` tương ứng với dữ liệu tên và điểm của các học sinh trong lớp. Chương trình sau ghi những thông tin này ra tệp `Data.out` có định dạng tương tự như `Data.inp`.

```
1 fname = "Data.out"
2 TenHS = ['Hà', 'Bình', 'Quang']
3 DiemHS = [9.6, 8.5, 7.2]
4
5 f = open(fname, "w", encoding = "UTF-8")
6 for i in range(len(TenHS)):
7     print(TenHS[i], DiemHS[i], file = f)
8 f.close()
```



Nhiệm vụ 2. Viết chương trình tra cứu điểm thi

Yêu cầu: Viết chương trình tra cứu điểm thi theo tên các học sinh trong lớp. Chương trình cho phép người dùng nhập tên của học sinh cần tra cứu, sau đó kiểm tra và thông báo điểm số của học sinh cần tìm.

Nhập dữ liệu: Yêu cầu người dùng nhập dữ liệu điểm học sinh từ tệp trên máy tính. Tệp bao gồm nhiều hàng, mỗi hàng gồm tên học sinh và điểm cách nhau bởi dấu cách. Ví dụ tệp có cấu trúc như Hình 20.1.

diem.inp	
Nam	7.8
Sơn	5.6
Hương	8.9
Huyền	7.4
Hà	9.5
Hùng	8.4

Hình 20.1

Tra cứu dữ liệu: Cho phép người dùng nhập tên học sinh cần tra cứu, tìm và in ra màn hình kết quả điểm thi của học sinh đó. Nếu không tìm thấy tên học sinh trong danh sách đã nhập, thông báo "không tìm thấy dữ liệu của học sinh".

Hướng dẫn:

Phân tích bài toán: Nhiệm vụ này có thể được thực hiện thông qua thuật toán tìm kiếm. Với dữ liệu đầu vào là tên của học sinh, chương trình cần sử dụng thuật toán tìm kiếm để kiểm tra xem tên của học sinh có tồn tại trong danh sách không. Nếu tồn tại thì học sinh cần tìm nằm ở vị trí nào trong danh sách. Do tên học sinh và điểm số được nhập vào danh sách theo cùng một thứ tự, từ vị trí tên học sinh trong danh sách tên, chúng ta có thể suy ra điểm số của học sinh đó trong danh sách điểm.

Chương trình có thể như sau:

Tracuuten.py

```
1 input_file = open("diem.inp", encoding="utf8") #encoding = utf8 để đọc được tiếng Việt
```

```

2 ten_list = []
3 diem_list = []
4 for line in input_file.readlines():# đọc lần lượt từng dòng trong tệp
5     ten,diem =line.split() # tách tên và điểm ở từng dòng
6     ten_list.append(ten)
7     diem_list.append(float(diem))
8 ten=''
9 while(ten!='end'):
10     ten = input('Nhập tên học sinh cần tra cứu, nhập từ khoá end để kết thúc:')
11     if (ten!='end'):
12         vitri=-1
13         for i in range(0,len(ten_list)):
14             if ten==ten_list[i]: # kiểm tra xem tên có tồn tại ở vị trí i
15                 vitri=i
16         if vitri==-1:
17             print('Không tìm thấy học sinh trong danh sách')
18         else:
19             print('Điểm số của học sinh là:',diem_list[vitri])

```



Nhiệm vụ 3. Viết chương trình kiểm tra điểm thi

Yêu cầu: Viết chương trình kiểm tra điểm thi của các học sinh trong một lớp học. Điểm thi của học sinh được ghi trong tệp `diemthi_sx.inp`, trong đó mỗi điểm thi của các học sinh được viết trong một hàng và được sắp xếp theo thứ tự tăng dần. Chương trình đọc dữ liệu điểm thi từ tệp, sau đó cho phép người dùng nhập một điểm số cần kiểm tra. Nếu điểm số có tồn tại thì in ra vị trí mà điểm số đó xuất hiện trong tệp, nếu điểm số không tồn tại thì in ra thông báo điểm số không tồn tại. Ví dụ tệp có cấu trúc như Hình 20.2.

<code>diemthi_sx.inp</code>
5.6
5.8
6.8
7.4
7.5
7.9

Hình 20.2

Hướng dẫn:

Phân tích bài toán: Tương tự như Nhiệm vụ 2, nhiệm vụ này vẫn được thực hiện bằng cách sử dụng thuật toán tìm kiếm. Trong nhiệm vụ này, dữ liệu điểm số đã được sắp xếp theo thứ tự tăng dần do đó chúng ta có thể áp dụng thuật toán tìm kiếm nhị phân để gia tăng tốc độ tìm kiếm.

Chương trình có thể như sau:

`Kiemtradiem_sx.py`

```

1 def BinarySearch(A,K):
2     left = 0
3     right = len(A) - 1
4     while left <= right:
5         mid = (left + right)//2
6         if A[mid] == K:
7             return mid
8         elif A[mid] < K:

```

```

9         left = mid + 1
10    else:
11        right = mid - 1
12    return -1
13 input_file = open("diemthi_sx.inp") # đọc dữ liệu từ file diemthi_sx.inp
14 ds_diem = [] #khởi tạo danh sách để chứa điểm số đọc từ file
15 for line in input_file.readlines():
16     ds_diem.append(float(line)) # lần lượt đọc các dòng của file và đưa vào
    danh sách điểm thi
17 input_file.close()
18 diem = float(input('Nhập điểm số cần kiểm tra:'))
19 vitri = BinarySearch(ds_diem, diem) # gọi đến hàm BinarySearch kiểm tra xem
    có phần tử 'diem' trong danh sách 'ds_diem' hay không
20 if vitri == -1:
21     print('Không tồn tại điểm số cần tìm trong danh sách')
22 else:
23     print('Điểm cần tìm nằm ở hàng thứ', vitri, 'trong danh sách')

```



LUYỆN TẬP

Chỉnh sửa lại chương trình của Nhiệm vụ 3 để cho phép chương trình có thể tìm kiếm điểm số trên danh sách điểm số được sắp xếp theo thứ tự giảm dần.



VẬN DỤNG

Viết chương trình tra cứu tên theo điểm thi của học sinh trong lớp. Chương trình cho phép người dùng nhập vào khoảng điểm số cần tìm kiếm (ví dụ từ 6 đến 8). Chương trình kiểm tra và thông báo tên của học sinh có điểm số nằm trong khoảng tương ứng. Giải bài toán trong hai trường hợp: điểm được sắp xếp theo thứ tự ngẫu nhiên như trong Nhiệm vụ 1 hoặc điểm được sắp xếp theo thứ tự tăng dần như sau:

Sơn 5.6
 Huyền 7.4
 Nam 7.8
 Hùng 8.4
 Hương 8.9
 Hà 9.5