README
Bài tập lớn

môn

Trí Tuệ Nhân Tạo

Nhóm 4

Cấn Duy Cát
Nguyễn Minh Trang
Nguyễn Mạnh Duy
Kiều Minh Đức
Nguyễn Mạnh Cường

01-04-2016

Mục lục

Lịch sử thay đổi

Người thay đổi	Ngày thay đổi	Lý do thay đổi	Phiên bản
Cat Can	01-04-2016	Tạo tài liệu	1.0
Duy Nguyen	08-04-2016	Viết sơ bộ tài liệu	1.1
Cat Can	13-04-2016	Hoàn thành tài liệu	1.2

1. Cấu trúc mã nguồn

Cấu trúc mã nguồn gồm 2 module chính: **ai_lib** và **algorithms**. Ngoài ra mã nguồn còn bao gồm 1 module **generate_input** độc lập để sinh toàn bộ input có đáp án, file main.py kèm với file input result để chạy thử thuật toán.

1.1 Module ai_lib

Module **ai_lib** chứa toàn bộ các thư viện có sẵn và được viết mới phục vụ cho việc thực hiện thuật toán tìm kiếm.

- **3_letters_dictionary**: từ điển các từ gồm 3 chữ cái được cung cấp sẵn.
- **bigram_frequence_list**: danh sách thống kê tần suất đứng cạnh nhau của 2 chữ cái được cung cấp sẵn.
- **inputs_dict**: từ điển toàn bộ 12431 bộ kết quả có thể có, được nhóm tạo ra bằng thuật toán **gen_input** được thực hiện trong module **generate_input**. Các bộ kết quả được sắp xếp theo thứ tự alphabet.
- **ai_helper.py**: gồm các hàm hỗ trợ cho việc chạy thuật toán, gồm: chuyển từ mảng 1 chiều sang mảng 2 chiều và ngược lại, lấy dữ liệu tần suất.
- **ai_io.py**: chứa các hàm vào và ra, được dùng để đọc và ghi dữ liệu.

1.2 Module algorithms

Module algorithms chứa toàn bộ các thuật toán tìm kiếm. Gồm 2 thuật toán **algorithm_part_2** là thuật toán theo yêu cầu, và thuật toán **algorithm_part_3** là thuật toán được tối ưu hàm đánh giá.

Mỗi hàm đánh giá được định nghĩa là 1 class, có 2 thuộc tính là **inputs** và **results**. Hai thuộc tính trên là 2 danh sách các input và kết quả tương ứng.

Interface để viết các hàm như sau:

- **Hàm khởi tạo**: hàm khởi tạo nhận tham số là một danh sách các inputs, mỗi input là một string gồm 9 ký tự.
- **Hàm heuristic**: hàm heuristic nhận tham số là một trạng thái và thứ tự của bộ input đang được chạy. Hàm heuristic trả về giá trị đánh giá của trạng thái đó, giá trị cao hơn là tốt hơn.
- **Hàm execute**: hàm execute nhận tham số là 2 cài đặt trace và pause. Hàm execute sẽ thực hiện hàm tìm kiếm trên toàn bộ các bộ inputs và lưu kết quả vào results. Nếu có kết quả thì kết quả ghi vào là một tuple (kết quả, số state đã duyệt, thời gian chạy). Trong trường hợp không có kết quả thì kết quả ghi vào là None.

1.3 Module generate_input

Gồm hàm để sinh ra các bộ kết quả chuẩn từ file từ điển được cung cấp trong module **ai_lib**. Kết quả của thuật toán sinh kết quả được ghi trong file **inputs.txt** và sẽ được dùng để là từ điển đầu vào trong thư viện **ai_lib**.

1.4 Các file khác

- **main.py**: đây là file được viết để test thuật toán đã được viết. Các bước để test thuật toán sẽ được mô tả chi tiết trong các phần sau của tài liệu.
- **input**: đây là file đầu vào của main.py. File gồm các bộ input là string gồm 9 ký tự, mỗi bộ input được ghi trên một dòng.
  File input có thể được cập nhật nếu người dùng tạo mới bộ input trong hàm main hoặc được lưu lại để chạy lần hai để so sánh hai thuật toán.
- **result**: các file được sinh ra sau khi chạy thuật toán, chứa kết quả tương ứng với các inputs trong file đầu vào.

2. Input và Output

Input và output của thuật toán đều là string có độ dài 9 ký tự. Output được hiểu là 3 ký tự đầu nằm trên dòng đầu tiên, 3 ký tự tiếp theo nằm trên dòng thứ hai và 3 ký tự cuối cùng nằm trên dòng cuối.

3. Cách chạy thuật toán

Để chạy thuật toán cần thực hiện các bước:

- **Bước 1**: mở command line hoặc terminal, tìm đến thư mục SourceCode
- **Bước 2**: chạy lệnh "**python main.py**"
- **Bước 3**: chọn có(y) hoặc không(n) tạo bộ input mới, mặc định là có(y).
  Nếu có tạo bộ input mới thì nhập tiếp số lượng bộ input muốn tạo. Chương trình tự động sinh inputs và lưu vào file "input"
- **Bước 4**: lựa chọn thuật toán **algorithm_part_2**(1) hoặc **algorithm_part_3**(2) hoặc chạy cả hai thuật toán(3).
- **Bước 5**: chọn cài đặt có(y) hoặc không(n) lần vết, mặc định là không(n).
  Nếu lựa chọn là có lần vết thì sẽ chọn thêm có(y) hay không(n) thực hiện từng bước, mặc định là có(y). Việc thực hiện từng bước sẽ dễ dàng quan sát được thuật toán thực hiện như thế nào?
- **Bước 6**: sau khi thuật toán chạy xong nhập kết quả sẽ được ghi ra file result. Màn hình hiển thị số input có kết quả và thời gian, số state đã duyệt trung bình.

4. Lưu ý

Việc lựa chọn ngẫu nhiên 9 ký tự trong 26 chữ cái của bảng chữ cái tiếng Anh tương đương với việc chọn một phần tử trong số chỉnh hợp lặp chập 9 của 26, tương đương 5.4 nghìn tỷ cách lựa chọn, tuy nhiên số bộ input có kết quả chỉ vào khoảng 3 tỷ bộ.

Do đó việc sinh ngẫu nhiên bộ input là không khả thi nên việc sinh input sẽ được thực hiện bằng cách lựa chọn ngẫu nhiên trong từ điển kết quả nhóm đã sinh ra sau đó trộn ngẫu nhiên các ký tự trong chuỗi 9 ký tự. Việc sinh input được thực hiện độc lập và ngẫu nhiên so với việc chạy thuật toán nên kết quả thuật toán chạy được là hoàn toàn khách quan.

5. Cách thức thực hiện

Để thực hiện bài toán, nhóm đã thực hiện qua các giai đoạn:

- Viết cấu trúc của project gồm các thư viện vào ra.
- Thực hiện tạo các bộ inputs và outputs mẫu.
- Viết interface cho class thuật toán gồm các hàm: constructor, heuristic, execute.
- Triển khai các thuật toán theo yêu cầu.
- Làm mịn thuật toán phần 3 bằng cách bổ sung các hàm cộng và trừ điểm cho hàm đánh giá.
- Thực hiện chạy thử trên các bộ inputs và outputs mẫu đã tạo.
- Viết hàm main để đánh giá thuật toán.
- Viết báo cáo.