**알고리즘 기말과제**

**융합전자공학부 2015003654 이호영**

* 구현 환경 및 인터페이스

Visual Studio 2019에서 컴파일하였습니다. 코드를 실행하면 input.txt에서 입력 받은 문자열을 라인 별로 다시 출력하고, 각 문자열의 길이를 출력하였습니다. 그 뒤 각 문자열끼리 겹치는 문자 개수를 20\*20 weight 배열에 저장하여서 출력하였고 겹치는 것이 가장 많은 순서대로 path를 형성하여 path를 출력하였습니다. Path 순서대로 다시 문자열들을 출력해서 겹치는 부분이 맞게 나왔는지 확인했습니다. 그리고 마지막에 superstring을 출력하였습니다.

* 아이디어 설명

각 문자열을 그래프의 한 노드로 보고, 그래프의 edge를 문자열끼리 겹치는 문자 개수로 잡았습니다. 그래프는 weight[i][j]의 2차원 배열로 나타내었고, i번째 문자열이 앞에, j번째 문자열이 뒤에 와서 이어지는 것으로 보았습니다. 그래서 그래프는 directed graph가 되었습니다.

Shortest common superstring problem은 traveling salesman problem으로 reduce 가능하다는 아이디어를 얻고, 제가 구현한 그래프에서 overlap이 최대로 일어나는 경로를 찾으면 그게 제가 찾을 수 있는 최적의 superstring이 될 것이라 생각했습니다. 경로 찾기 문제는 최소 비용을 찾기를 원하기에 그래프의 가중치에 -1을 곱하여서 똑같이 최소 비용 경로를 찾는 아이디어를 사용했습니다. Tsp은 branch and bound 방식으로 가중치 배열을 계속 업데이트해가면서 least cost bound를 구하며 풀어나갔습니다. 그래서 overlap이 크게 나오는 최적의 경로를 출력한 뒤엔 그 순서대로 문자열을 출력하려 했습니다. 하지만 겹치는 부분을 한번만 출력하는 부분을 해결하기가 어려웠고, 고민 끝에 문자 하나 하나씩 출력을 하고 배열의 인덱스는 전체 길이에서 겹치는 부분의 길이를 빼고 출력하였습니다.

* 참조한 source

Overlap 개수를 구하는 함수를 작성할 때 예전 pattern matching에서 구현하였던 code가 생각나서 그때 code를 참조하였고 overlap을 찾는 부분에서 고전을 하다가 제출일이었던 화요일에 친구에게 최후의 수단으로 goto문을 사용해보라는 의견을 듣고 goto 문을 사용하여서 overlap 개수를 구했습니다.

main.c

int overlapCount(char\* str1, char\* str2)

문자열 2개의 overlap 개수를 구하는 것으로, str1의 앞 부분과 str2의 뒷 부분이 overlap이 생길 때 overlap의 개수를 반환하는 함수입니다.

int branch\_and\_bound(int matrix[LINES][LINES], int total\_cost, int current)

matrix는 가중치를 나타낸 배열이고, total\_cost는 현재 노드까지의 총 cost이다. Current는 branch bound를 실행하는 노드의 index입니다. 그리고 더 이상 진행할 수 없을 때 least cost bound를 반환하고 그것이 total cost가 됩니다.