

## whw5

B911012 금예인

기존 backend.c 코드의 'whw5()'만 수정하였다. ladders.c, backend.h 역시 수정하지 않았고, 따라서 whw5()만 첨부하였다.

```
/****** Written Homework 5 *****/
```

```
void whw5()
{
    int idx = 0;
    int degree = 0;
    struct node* p = adj_list[idx];
    char hello[5] = { 'h','e','l','l','o' };
    char graph[5] = { 'g','r','a','p','h' };
    int arr_degree[N - 1] = { 0 };
    int i = 0;
    int max_d = 0;
    int sum_d = 0;
    double ave_d = 0;
    int num = 0;

    //1번
    printf("\n1. \n");
    printf("(a) : ");
    idx = search_index(hello);
    for (i = 0; i < N; i++) {
        if (adj_mat[idx][i] == 1) {
            print_word(i);
            printf(" ");
            degree++;
        }
    }
    printf("\nThe degree of hello is %d\n", degree);

    printf("(b) : ");
    degree = 0;
    idx = search_index(graph);
    p = adj_list[idx];
    for (i = 0; i < N; i++) {
        if (adj_mat[idx][i] == 1) {
            print_word(i);
            printf(" ");
        }
    }
}
```

```

        degree++;
    }
}
printf("\nThe degree of graph is %d\n\n", degree);

```

```

//2번
printf("2. \n");
idx = 0;
for (idx = 0; idx < N; idx++) {
    p = adj_list[idx];
    degree = 0;
    for (i = 0; i < N; i++) {
        if (adj_mat[idx][i] == 1) {
            degree++;
        }
    }
    arr_degree[degree]++;
}
for (max_d = 0; max_d < N - 1; max_d++) {
    num += arr_degree[max_d];
    if (num == N) {
        break;
    }
}
for (i = 0; i <= max_d; i++) {
    printf("%d: %d\n", i, arr_degree[i]);
}
printf("\n");

```

```

//3번
printf("3. ");
printf("The maximum degree is %d\n\n", max_d);

```

```

//4번
printf("4. ");
idx = 0;
for (idx = 0; idx < N; idx++) {
    degree = 0;
    for (i = 0; i < N; i++) {
        if (adj_mat[idx][i] == 1) {
            degree++;
        }
    }
}

```

```

        if (degree == max_d) {
            print_word(idx);
            printf(" ");
        }
    }
    printf("\n\n");

    //5번
    printf("5. ");
    for (i = 0; i < N - 1; i++) {
        sum_d += i * arr_degree[i];
    }
    ave_d = (double)sum_d / N;
    printf("The average degree is %f\n\n", ave_d);

    //6번
    printf("6. ");
    printf("The number of nodes are %d\n\n", sum_d);

    //7번
    printf("7. ");
    printf("The minimum possible size required of POOL SIZE in backend.c is %d\n\n",
sum_d);
    // You may use this space for your homework :)
}

```

실행 결과는 다음과 같다.

```
[B911012@localhost whw5]$ ladders
All the words were successfully read!
Adjacency matrix was successfully constructed!
Adjacency list was successfully constructed!

1.
(a) : cello hallo hells hullo jello
The degree of hello is 5
(b) : grape grapy
The degree of graph is 2

2.
0: 671
1: 774
2: 727
3: 638
4: 523
5: 428
6: 329
7: 280
8: 249
9: 213
10: 188
11: 162
12: 120
13: 116
14: 102
15: 75
16: 53
17: 32
18: 32
19: 20
20: 8
21: 6
22: 4
23: 2
24: 3
25: 2

3. The maximum degree is 25

4. bares cores

5. The average degree is 4.910544

6. The number of nodes are 28270

7. The minimum possible size required of POOL SIZE in backend.c is 28270

Welcome to Ladders!

Starting word: █
```

hello와 adjacent한 단어는 cello, hallo, hells, hullo, jello로, hello의 degree는 5였다.

graph와 adjacent한 단어는 grape, grapy로, graph의 degree는 2였다.

degree에 대한 table을 작성해본 결과, 최소 degree는 0이었고, 이에 해당하는 단어는 671개였다. 최대 degree는 25였으며, 이에 해당하는 단어는 bares, cores로 2개였다.

degree의 평균은

$$(0 * (\text{degree가 0인 단어 수}) + 1 * (\text{degree가 1인 단어 수}) + \dots + 25 * (\text{degree가 25인 단어 수})) / N$$
로 구할 수 있고, 값은 4.910544였다.

our adjacency list에는 총 28270개의 노드가 있었고, POOL\_SIZE로 가능한 가장 작은 값 역시 28270이다.