WHW4

B911061 노현근

```
****Question 1****
hello's degree:
cello
hallo
hells
hullo
jello

graph's degree:
grape
grapy

****Question 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
26 : 0
27 : 0
28 : 0
29 : 0
****Question 3****
max_degree: 25

****Question 4****
max_degree_word: bares
max_degree_word: cores

****Question 5****
average_degree: 4.910544

****Question 6****
adjacency list's total node: 28270

****Question 7****
mininum possible size: 28270
```

Answer

1번.

(a)
```
hello
cello
hallo
hells
hullo
jello
```

(b)
```
grape
grapy
```

2번.

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

3번.

Maximum degree : 25

4번.

Words that have the maximum degree: `bares`
`cores`

5번.

Average degree: 4.910544

6번.

Total node that our adjacency list have: 28270

7번.

Minimum possible size required of `POOL_SIZE` in `backend.c`: 28270

Code

```
void whw5()
{
	int word_index;
	int i, j;
	int array[30] = { 0,};

	/****Question 1****/
	printf("****Question 1****\n");
	printf("hello's degree: \n");
	word_index = search_index("hello");
	for (i = 0; i < 5757; i++)
		if (adj_mat[word_index][i] == 1) {
			print_word(i);
			printf("\n");
		}
	printf("\ngraph's degree: \n");
	word_index = search_index("graph");
	for (i = 0; i < 5757; i++)
		if (adj_mat[word_index][i] == 1) {
			print_word(i);
			printf("\n");
		}

	/****Question 2****/
	printf("\n****Question 2****\n");
	for (i = 0; i < 5757; i++) {
		int count = 0;
		for (j = 0; j < 5757; j++)
			if (adj_mat[i][j] == 1)
				count++;
		array[count]++;
	}
	for (i = 0; i < 30; i++)
		printf("%d : %d\n", i, array[i]);
```

```
/****Question 3****/
printf("\n****Question 3****\n");
int max_degree;

for (i = 0; i < 30; i++)
	if (array[i] != 0)
		max_degree = i;
printf("max_degree: %d\n", max_degree);

/****Question 4****/
printf("\n****Question 4****\n");
for (i = 0; i < 5757; i++) {
	int count = 0;
	for (j = 0; j < 5757; j++)
		if (adj_mat[i][j] == 1)
			count++;
	if (count == 25) {
		printf("max_degree_word: ");
		print_word(i);
		printf("\n");
	}
}

/****Question 5****/
printf("\n****Question 5****\n");
int sum;

sum = 0;
for (i = 0; i < 30; i++)
	sum += array[i] * i;
printf("average_degree: %lf\n", (double)sum / (double)5757);
```

```
	/****Question 6****/
	printf("\n****Question 6****\n");
	sum = 0;
	int check = 0;
	for (i = 0; i < 5757; i++) {
		struct node * r;
		int count;

		count = 0;
		r = adj_list[i];
		while (r) {
			count++;
			r = r->next;
		}
		if (count == 1)
			check++;
		sum += count;
	}
	printf("adjacency list's total node: %d\n", sum);

	/****Question 7****/
	printf("\n****Question 7****\n");
	printf("mininum possible size: %d\n", sum);
}
```