과제명	Written Homework 4
담당교수	배성일
분반	2
학번	B617065
이름	신성우

1. Solution

```
==================== Question 1-(a) ====================
All the words adjacent to hello: cello hallo hells hullo jello
The degree of hello: 5


==================== Question 1-(b) ====================
All the words adjacent to hello: grape grapy
The degree of graph: 2


==================== Question 2 ====================
The table of distribution of degrees
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


==================== Question 3 ====================
The maximum degree: 25


==================== Question 4 ====================
bares cores


==================== Question 5 ====================
The average degree: 4.910544


==================== Question 6 ====================
Our adjacency list has 28270 nodes
```

- Question 7

  minimum possible size required of POOL SIZE in backend.c는 adjacency_list의 노드의 총 개수인 28270이다.

2. Source code

```
/********** Written Homework 5 **********/
void  whw5(void)
{
  // Question 1.
  printf("==================== Question 1-(a) ====================\n");
  int         idx;
  int         deg;
  struct node *cur;

  idx = search_index("hello");
  cur = adj_list[idx];
  deg = 0;
  printf("All the words adjacent to hello: ");
  while (cur)
  {
    print_word(cur->index);
    printf(" ");
    ++deg;
```

```
  cur = cur->next;
}
printf("\nThe degree of hello: %d\n", deg);

printf("\n\n==================== Question 1-(b) ====================\n");
idx = search_index("graph");
cur = adj_list[idx];
deg = 0;
printf("All the words adjacent to hello: ");
while (cur)
{
  print_word(cur->index);
  printf(" ");
  ++deg;
  cur = cur->next;
}
printf("\nThe degree of graph: %d\n", deg);

// Question 2.
printf("\n\n==================== Question 2 ====================\n");
int map_freq[50] = {0, };
int map_deg[N] = {0, };
int i;
int freq;
int max_deg;

i = 0;
while (i < N)
{
  cur = adj_list[i];
  freq = 0;
  while (cur)
  {
    ++freq;
    cur = cur->next;
  }
  ++map_freq[freq];
  map_deg[i] = freq;
  ++i;
}
i = 0;
printf("The table of distribution of degrees\n");
while (i < 50)
{
  if (map_freq[i] != 0)
  {
    printf("%d:\t%d\n", i, map_freq[i]);
    max_deg = i;
  }
```

```
    ++i;
  }

  // Question 3.
  printf("\n\n==================== Question 3 ====================\n");
  printf("The maximum degree: %d\n", max_deg);

  // Question 4.
  printf("\n\n==================== Question 4 ====================\n");
  i = 0;
  while (i < N)
  {
    if (map_deg[i] == max_deg)
    {
      print_word(i);
      printf(" ");
    }
    printf("\n")
    ++i;
  }

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

  i = 0;
  sum = 0;
  while (i < 50)
  {
    sum += i * map_freq[i];
    ++i;
  }
  printf("The average degree: %f\n", (float)sum / N);

  // Question 6.
  printf("\n\n==================== Question 6 ====================\n");
  sum = 0;
  i = 0;
  while (i < N)
  {
    sum += map_deg[i];
    ++i;
  }
  printf("Our adjacency list has %d nodes\n", sum);
}
```