

Homework2

2017141493004 常家奇

- 4.5

1. 可以完成操作.新建一个节点,将其内容赋值为当前节点,同时将原节点指向新节点,同时更新end即可.更复杂.
2. 除了最后一个节点,其余均可在 $O(1)$ 内完成.不能连续删除.

- 4.10

D是数组的长度,E是一个元素所占空间(int,byte..),P是一个指针所占空间,n是链表中元素个数.
当E和P单位统一为byte即可.

- 4.13

```
#include "Stack.cpp"
```

```
template <typename E> class ATStack: public Stack<E> {
private:
    int max_size;
    int top_one;
    int top_two;
    E *list_array;

public:
    ATStack(int size = default_size) { max_size = size; top_one = 0; top_two = 0; list_array = r

    ~ATStack() { delete[] list_array; }
    void clear() { top_one = 0; top_two=0; }
    void push_first_stack(const E& v) {
        Assert( top_one + top_two < max_size, "Stack is full");
        list_array[top_one++] = v;
    }
    void push_second_stack(const E& v) {
        Assert( top_one + top_two < max_size, "Stack is full");
        list_array[max_size-1-top_two++] = v;
    }
    E pop_first_stack() {
        Assert(top_one!=0, "Stack is empty");
        return list_array[--top];
    }
    E pop_second_stack() {
        Assert(top_two!=0, "Stack is empty");
        return list_array[max_size - 1 - --top];
    }
    const E& top_first() const {
        Assert(top_one!=0, "Stack is empty");
        return list_array[top_one-1];
    }
    const E& top_second() const {
        Assert(top_two!=0, "Stack is empty");
        return list_array[max_size-1-top_two];
    }
    int length_first() const { return top_one; }
    int length_second() const { return top_two; }
}
```

- 4.18

```
void rev_q(const Queue& q, const Stack& s){
    while (q.length()) s.push(q.dequeue());
    while (s.length()) q.enqueue(s.pop());
}
```

- 补充作业题

- 单项选择题

1. C
2. D
3. D
4. A
5. D
6. B
7. B
8. D
9. A
10. A

- 综合题

1.
$$x = \frac{\sum_{i=0}^n i}{n+1} = \frac{n}{2}$$

2. *CDEBA, CDBEA, CDBAE*

3. 第一个不能，第二个可以。

输出的第一个元素为d，那么只能将abcd全部压栈再弹出。第二个元素为b，与此时栈顶元素c不符。

顺序： push(a) push(b) push(c) pop() pop() push(d) pop() pop()

4. 假设只有三个元素i,j,k, 且*i<j<k*,它们出栈的顺序只有5种: *kji, jki, jik, ijk, ikj*没有一种顺序满足输出序列，所以假设在三个元素下成立。

因为对于任意相邻的三个元素均满足此条件，所以推广到任选三个元素依旧成立。证毕。