

Nguyễn Minh Đức

1712358

Lý thuyết Cấu trúc dữ liệu và giải thuật

Bài tập tuần 5

Phần trắc nghiệm

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

Phần tự luận

1. Trừu tượng là sự khái quát hóa trong tư duy từ các thuộc tính, quan hệ của sự vật.

Ví dụ như tình yêu, sự tự do, sự phẫn nộ,... đó là những thứ không thể nhìn thấy, không thể nghe, sờ hay cầm, nắm,... nhưng lại được định nghĩa chính xác trong tâm trí con người, được suy ra từ các tính chất của nó, các quan hệ của nó.

2. EAS*Y**Q***ST***I*ON

kết quả khi in ra màn hình

SYAQETSI

3. A=5, B=3, C=7

a. Tạo ngăn xếp

NULL

b. Push A

5

c. Push C*C

5 49

d. Pop và lưu trữ vào B

5

B = 49

e. Push B+A

5 54

f. Pop và lưu trữ vào A

5

A = 54

g. Pop và lưu trữ vào B

NULL

B = 5

4. $S = "5 + ((1 + 2) * 4) - 3"$

Ta đi từ trái qua phải

Gặp số 5 đầu tiên → đưa vào ngăn xếp

Sau đó gặp dấu cộng, v là 5 cộng với 1 số, số đó bằng $((1 + 2) * 4)$ nên ta sẽ phân tích phần $((1 + 2) * 4)$ trước, cũng đi từ trái qua phải thực hiện phép $1+2$ trước

→ đưa vào ngăn xếp $1\ 2\ +$, sau đó kết quả này nhân với 4

→ đưa vào ngăn xếp $1\ 2\ +\ 4\ *$

Rồi kết quả này cộng với 5

→ $5\ 1\ 2\ +\ 4\ *\ +$

Rồi kết quả này trừ đi 3

→ $5\ 1\ 2\ +\ 4\ *\ +\ 3\ -$

5. Ban đầu:

	A	B	C		
--	---	---	---	--	--

5.1 Bổ sung E vào hàng đợi

E	A	B	C		
---	---	---	---	--	--

5.2 Loại 2 phần tử ra khỏi hàng đợi

E	A				
---	---	--	--	--	--

5.3 Bổ sung I, J, K vào hàng đợi

E	A		K	J	I
---	---	--	---	---	---

5.4 Loại 2 phần tử khỏi hàng đợi

			K	J	I
--	--	--	---	---	---

5.5 Bổ sung O vào hàng đợi

		O	K	I	J
--	--	---	---	---	---

5.6 Loại 2 phần tử khỏi hàng đợi

		O	K		
--	--	---	---	--	--

7. Với S1, S2 là 2 danh sách liên kết, vậy để trả về S chứa tất cả dữ liệu của S1 và S2 ta chỉ việc nối S1, S2 lại với nhau tạo ra danh liên kết mới, có thể nối đầu S2 vào đuôi S1. Thời gian chỉ tốn $O(1)$

8. Ta có thể đưa dữ liệu từ S qua S' để thực hiện ý đồ bài toán

a) Ta sẽ đưa toàn bộ dữ liệu qua S', rồi sau đó khi đưa ngược lại về S ta sẽ in ra màn hình

```
while(S.Empty())
    S'.Push(S.Pop());
while(S'.Empty())
{
    S.Push(S'.Pop());
    cout << S.Top();
}
```

b) Cũng làm như trên nhưng thay vì in ra thì ta đếm số phần tử

```
int n = 0;
while(S.Empty())
    S'.Push(S.Pop());
```

```

while(S'.Empty())
{
    S.Push(S'.Pop());
    n++;
}

```

c) Ta cũng làm như trên, nhưng tại mỗi bước ta sẽ kiểm tra xem phần tử hiện có bằng x không, nếu bằng thì ta không thêm ngược vào S

```

while(S.Empty())
    S'.Push(S.Pop()) ;
while(S'.Empty())
{
    if(S'.Top() != x)
        S.Push(S'.Pop());
    else
        S'.Pop();
}

```

9.

a - a
b - e
c - e
d - b
e - b
f - b
g - b

10.

```

num1 = 5 ; num2 = 1 ; num3 = 4 ;
aQueue.enqueue(num2) ; //1
aQueue.enqueue(num3) ; //1 4
aQueue.dequeue() ; //4
aQueue.enqueue(num1 - num2) ; // 4 4
num1 = aQueue.peek() ; //num1 = 4

```

```
aQueue.dequeue() ; // 4
num2 = aQueue.peek() ; //num2 = 4
aQueue.dequeue() ; // NULL
cout << num2 << " " << num1 << " " << num3 << endl ;
```

Vậy kết quả là $\text{num1} = 4$, $\text{num2} = 4$ và $\text{num3} = 4$