

# UTS Bootcamp Data Structure

## Stack and queue

1. Pertama, stack merupakan elemen yang insert paling akhir adalah elemen yang pertama keluar, sedangkan queue adalah kebalikkannya dimana elemen yang dimasukkan paling awal akan yang pertama keluar. Kedua, insert dan delete di stack akan terjadi dari ujung yang disebut top, sedangkan queue insert dan delete terjadi dari end yang berlawanan, sehingga insert terjadi di belakang dan delete terjadi dari depan. Dan ketiga untuk insert di stack disebut push dan untuk insert di queue disebut enqueue.
2. Prefix notation dilihat dari kata pre, sehingga operatornya ditulis sebelum operandnya. Untuk infix operatornya ditulis diantara operandnya, sedangkan untuk postfix dapat dilihat dari kata post, sehingga operatornya ditulis setelah operandnya.

### Contoh infix menggunakan stack:

Evaluasi infix:  $5 + 5 * (5 - 2) / 3$

$5 + 5 * (5 - 2) / 3$

$5 + 5 * 3 / 3$

$5 + 15 / 3$

$5 + 5$

10

**cari operator precedence yang paling tinggi, yaitu ()**

**cari operator precedence yang paling tinggi, yaitu \***

**cari operator precedence yang paling tinggi, yaitu /**

**cari operator precedence yang paling tinggi, yaitu +**

Untuk mencari operator precedence yang tertinggi dapat menggunakan looping dengan kompleksitas  $O(N^2)$  atau nested loop karena harus diloop terus menerus sampai angkanya atau operandnya habis. N itu adalah Panjang dari string dan angka di atas dapat dianggap sebagai string atau character.

Langkah-langkahnya:

1. Mencari operator precedence yang tertinggi.
2. Menggunakan looping sampai operandnya habis atau selesai.

### Contoh prefix menggunakan stack:

Evaluasi:  $+ 5 - x 6 5 ^ 3 2$

$+ 5 - x 6 5 ^ 3 2$

$+ 5 - x 6 5 ^ 3 2$

$+ 5 - x 6 5 9$

$+ 5 - x 6 5 9$

$+ 5 - 30 9$

$+ 5 - 30 9$

$+ 5 21$

$+ 5 21$

28

**scan sampai menemukan operator pertama**

**mengitung  $3 ^ 2$**

**scan lagi sampai menemukan operator berikutnya**

**menghitung  $6 * 5$**

**scan lagi sampai menemukan operator selanjutnya**

**menghitung  $30 - 9$**

**scan lagi sampai menemukan operator selanjutnya**

**menghitung  $5 + 21$**

Langkah-langkahnya:

1. Scan prefix dari kanan ke kiri.
2. Jika berupa operand atau angka, maka masuk ke stack.
3. Jika berupa operator, maka pop stack 2 kali kemudian menghitung hasil dan push ke dalam stack
4. Looping sampai ujung kiri prefixnya.

**Contoh postfix menggunakan stack:**

Evaluasi: 5 6 5 x 3 2 ^ - +

5 6 5 x 3 2 ^ - +	scan sampai menemukan operator pertama
5 <u>6 5</u> x 3 2 ^ - +	menghitung 6 * 5
5 30 3 2 ^ - +	scan lagi untuk mencari operator selanjutnya
5 30 <u>3 2</u> ^ - +	menghitung 3 ^ 2
5 30 9 - +	scan lagi untuk mencari operator berikutnya
5 <u>30 9</u> - +	menghitung 30 - 9
5 21 +	scan lagi untuk mencari operator selanjutnya
<u>5 21</u> +	menghitung 5 + 21
26	

Langkah-langkahnya:

5. Scan prefix dari kiri ke kanan.
6. Jika berupa operand atau angka, maka masuk ke stack.
7. Jika berupa operator, maka pop stack 2 kali kemudian menghitung hasil dan push ke dalam stack
8. Looping sampai ujung kanan postfixnya.