

AD:SELENAY

SOYAD:HUR

ÖĞRENCİ NUMARASI: **23080103374**

ÖĞRETİM GÖREVLİSİ:Dr. Öğr. Üyesi SELMAN HIZAL

Rekürsif ve İteratif Çözümün Karşılaştırılması

- Rekürsif Çözüm(Özyineleme)

Rekürsif çözüm, Hanoi Kuleleri probleminin tanımına doğrudan bir yaklaşımdır. Şu adımları takip eder:

1) $n-1$ diskini kaynak çubuktan yardımcı çubuğa taşır.

2) n .diski kaynak çubuktan hedef çubuğa taşır.

3) $n-1$ diskini yardımcı çubuktan hedef çubuğa taşır.

```
void Hanoi(int n, char kaynak, char ara, char hedef) {  
    if (n == 1) {  
        diskTasi(1, kaynak, hedef);  
    } else {  
        Hanoi(n - 1, kaynak, hedef, ara);  
        diskTasi(n, kaynak, hedef);  
        Hanoi(n - 1, ara, kaynak, hedef);  
    }  
}
```

- Avantajlar:

Rekürsif çözüm, problemin tanımına doğrudan yaklaşır ve oldukça basit ve okunabilirliği daha yüksektir

Dezavantajlar:

Çok sayıda disk için,stack overflow(yığın taşması) meydana gelebilir

- İteratif Çözüm(Döngüsel)

Rekürsif çağrılarını simüle etmek için bir yığın kullanır. Bu yöntem, kendi yığın veri yapısını yöneterek stack overflow riskini ortadan kaldırır.

```

void HanoiIteratif(int n, char kaynak, char ara, char hedef) {
    stack<tuple<int, char, char, char>> hareketler;
    hareketler.push(make_tuple(n, kaynak, ara, hedef));

    while (!hareketler.empty()) {
        int diskSayisi;
        char k, a, h;
        tie(diskSayisi, k, a, h) = hareketler.top();
        hareketler.pop();

        if (diskSayisi == 1) {
            diskTasi(1, k, h);
        } else {
            hareketler.push(make_tuple(diskSayisi - 1, a, k, h));
            hareketler.push(make_tuple(1, k, a, h));
            hareketler.push(make_tuple(diskSayisi - 1, k, h, a));
        }
    }
}

```

- Avantajlar:

Kendi yığın veri yapısını kullanarak, rekürsif çağrılarının neden olabileceği yığın taşması riskini ortadan kaldırır ve yığın kullanarak kontrol sağlar

- Dezavantajlar:

Kod, rekürsif çözümden daha karmaşıktır ve okunması zordur

- Performans karşılaştırması

```
cout << "Rekursif cozum suresi: " << sureRekursif.count() << " milisaniye" << endl;
```

```
cout << "Iteratif cozum suresi: " << sureIteratif.count() << " milisaniye" << endl;
```

- Sonuç:

- Rekürsif Çözüm: Basit ve doğal bir çözüm olup küçük disk sayıları için idealdir. Ancak büyük disk sayıları için yığın taşması riski taşır.

- İteratif Çözüm: Daha büyük veri setleri için güvenli ve etkili bir çözümdür. Ancak daha karmaşık ve okunması zordur.