**HANOİ KULELERİ ALGORİTMASI**

Mert Eren Keküç

Yazılım Mühendisliği 2. sınıf

Elimizde n tane disk olduğunu varsayalım.

1)Eğer n=1 olursa diskimizi bulunduğu yerden direkt hedefimize koyabiliriz.

2)Algoritmayı her n değer için genişletmek için recursive(özyinelemeli) fonksiyonları kullanacağız. Fonksiyona ilk giren toplam disk sayısı her bir adımda azaltılarak yapılacak olan tek hamleyi bulmalıyız.

3) hanoi(number,firstrod,twicerod,thirdrod) fonksiyonunda number kaç tane disk taşımak istediğimizin sayısıdır firstrod ilk çubuğumuzdur twicerod taşımak istediğimiz yer thirdrod ise yardımcı çubuktur.

4) hanoi(number - 1, firstrod, thirdrod, twicerod); --> bu kısımlarda taşıma işlemlerini yapıyoruz

hanoi(number - 1, thirdrod, twicerod, firstrod);

programımdaki bu kod parçaları 2 diskli hanoi kulesi için aşağıdaki gibi çalışmaktadır.(Hedef b diski)

hanoi(1,firstrod,thirdrod,twicerod) == 1.Disk A -> C

hanoi(1,firstrod,twicerod,thirdrod) == 2.Disk A -> B

hanoi(1,thirdrod,twicerod,firstrod) == 1.Disk C -> B

şeklinde çalışır. Yani burada 1 tane diskimiz önce yardımcı çubuğa taşınır sonra kalan tek diskimiz bitiş çubuğuna taşınır ardından yardımcı çubukta bulunan disk bitiş çubuğuna taşınır. 3 adımda kulemizi tamamlamış oluruz.

Ayrıca projemde süreyi tutabilmek için time.h kütüphanesinden faydalandım. clock\_t begin=clock(); ve clock\_t end=clock(); kod parçaları süreyi tutması gereken aralığı belirtiyor. printf("\t%d\t|\t%lf\t|\t%.0f",i,(double)(end-begin)/CLOCKS\_PER\_SEC,moves\_number); bu kod parçasında ise süreyi yazdırıyorum. Hamle sayısını da moves\_number = pow(2.00,i)-1; kod parçasında tutmuş oluyorum.

Aşağıdaki tablo bazı disk sayıları için değerler içermektedir.

|  |  |  |
| --- | --- | --- |
| DİSK SAYISI | SÜRE | HAMLE SAYISI |
| 1 | 0 | 1 |
| 10 | 0 | 1023 |
| 20 | 0.015000 | 1048575 |
| 25 | 0.234000 | 33554431 |
| 28 | 1.859000 | 268435455 |
| 29 | 3.609000 | 536870911 |
| 30 | 7.376000 | 1073741823 |
| 33 | 59.750000 | 8589934591 |