

No:

Ad Soyad:

İmza:

Cevaplarınızı soruların altlarındaki boşluklara yazınız. Ek kâğıt verilmeyecektir.

Süre: 75 dakika

Başarılar.

1- (25P)

Not1: Sadece sonucu yazmanız puan kazandırmaz. Sonuca nasıl ulaştığınızı göstermeniz gerekmektedir.

Not 2: $i=0:N$ toplam(a^i) = $(a^{N+1}-1)/(a-1)$

- a) Çalışma zamanı $T(n) = 3 \cdot T(n-(1/7)) + 1$ şeklinde verilen bir özyinelemeli fonksiyon için çalışma zamanını özyineleme ağacı kullanarak çözünüz. Ağaç yüksekliği (5P), çalışma zamanı (7P)

$$h=7n$$

$$T(n) = (3^{7n+1} - 1)/2$$

- b) $T(n) = (1/a) \cdot T(n-b) + 1$ formundaki bir özyineleme fonksiyonunun çalışma zamanının $3^{\log_2 n}$ (3 üzeri $\log_2 n$) olması için a (5P) ve b (8P) yi bulunuz.

$$a=1/3$$

$$b=n/2$$

2- (20P)

```
int main(void)
{
    srand( time( NULL ) );
    float say1=0; float say2=0; float s1,s2;
    int i,A,N;
    scanf("%d",&N); scanf("%d",&A);
    for (i=1;i<=N;i++)
    {
        s1 = rand( ) % A;
        s2 = rand( ) % A;
        if (s1*s1+s2*s2<A*A)
            say1=say1+1;
        if ((s1<A/2) && (s2<A/2))
            say2=say2+1;
    }
    printf("%f",say2/say1);
}
```

Yukarıdaki kod çalıştırıldığında ekrana çıkan sayı N'in büyük değerleri için yaklaşık olarak kaç'a yakınsar? Çözümünüzü bir şekil üzerinde gösteriniz.
İpucu: S1 ve S2'yi bir noktanın 2 koordinatı şeklinde düşünebilirsiniz.

```
say1 = (pi*A^2)/4
say2 = (A^2)/4
say2 / say1 = 1/pi
```

3- (30P)

Aşağıdaki sözde kod çalışmasını bitirdiğinde T (10P), A (10P) ve K (10P)'nin değerlerini N'e bağlı olarak bulunuz. N, 2'nin pozitif bir üssü olarak girilmektedir. (Ör: N=256, N=16 vb.)

```
oku(N)
T=0;
D=1;
while (D<N)
    K=1;
    while (K<=N)
        K=K+2;
        T=T+1;
    end
    D=D*2;
end

A=T;
D=0;
while (D<N)
    A=A-log2(N);
    D=D+2;
    K=K-1;
end

T=(N/2)*log2(N)
A=0
K=1+(N/2)
```

4- (25P)

6*6'lık bir satranç tahtasında (A), bir taş kullanıcının girdiği bir başlangıçtan (B1,B2) yine kullanıcının girdiği bir hedefe (H1,H2) rastgele hareketlerle gitmeye çalışmaktadır.

Taşın hedefe varmasının kaç harekete ve ne kadar maliyete sebep olduğunu bulan algoritmayı çizin.

Taş yukarı, aşağı, sağa, sola ve 4 çapraz olmak üzere 8 yöne hareket edebilir. Çapraz hareketler için maliyet 1.5, diğerleri için 1'dir.

Taşın tahtanın dışına çıkmamasını sağlamalısınız.

Rastgele hareket için 1-8 arası tamsayı üreten rand() fonksiyonu size verilmiştir.

Cözümünüzün geçerli olması için programda taş hareket ettirirken ve maliyeti hesaplarken "if / switch vb." kullanmayınız.

Olası bir durum: Gerçekte aynı yerden defalarca geçilebilir.

0 1 2 3 0 0

B 1 0 4 0 0

0 0 0 5 0 0

0 0 0 6 H 0

0 0 0 7 0 9

0 0 0 0 8 0 toplam 10 hareket, maliyet=7*1+3*1.5=11.5

Taş hareketleri için 3*8 lik bir matris oluşturun. İlk 2 satırı taş hareketlerini 3.

Satır maliyetlerini tutsun.

Mevcut konum hedef konum olmadığı sürece:

Rastgele sayı üret bu sayının gösterdiği hareket yapıldığında tahta dışına çıkılmıyorsa hareketi yap maliyeti arttır, tahta dışına çıkılıyorsa yeniden rastgele sayı üret.