

İleri Algoritma Analizi ve Tasarımı

2024-2025 Bahar Dönemi 1. Vize

Soru 1:

Aşağıdaki algoritma ve rekürans bağıntısı verilmiştir:

Algoritma $S(n)$:

```
if  $n \leq 1$ 
    return  $n$ 
else
    return  $S(n-1) + n*n$ 
```

1. Bu algoritmanın rekürans bağıntısını yazınız.
2. Bu algoritmanın karmaşıklık hesabını “backward substitution” yöntemi ile çözünüz.

Soru 2

Kod imzalama uygulamaları, yazılımları özel bir anahtarla dijital olarak imzalayarak, son kullanıcıların kodun güvenilir ve meşru bir kaynaktan geldiğini ve yayınlandığından beri değiştirilmediğini doğrulayabilmeyi sağlar. Bir kod imzalama uygulaması bu süreçte kriptolojik hash fonksiyonlarını kullanır. Kod imzalama sürecinde kullanılan hash fonksiyonlarının sahip olması beklenen özellikleri ve bu özelliklerin uygulama açısından önemini açıklayınız.

Soru 3

Bir matris üzerindeki gezinen bir robot için her bir hücre değeri o hücreye basabilmek için gerekli enerji tüketimini ifade etmektedir. Sol en üst hücreden başlandığında sağ en alt hücreye gidebilmek için harcanacak minimum enerji miktarını hesaplayan rekürsif c kodu aşağıda verilmiştir.

```
#include <stdio.h>
#include <stdlib.h>
#define INF 10000
int minPath(int i, int j, int E[][100]){
    int min;

    // Başlangıç noktasındaysak, o hücrenin değerini döndür
    if(i==0 && j == 0)
        return E[0][0];
```

```

// Eğer matris sınırlarını aştıysak, sonsuz değer döndür
if(i < 0 || j < 0)
    return INF;

// Yukarıdan ve soldan gelen minimum enerji yollarını hesapla
int from_top = minPath(i-1, j, E);
int from_left = minPath(i, j - 1, E);

// İki yoldan daha az enerji gerektireni seç
if(from_top < from_left)
    min = from_top;
else
    min = from_left;

// Mevcut hücrenin enerjisini ekleyip döndür
return min + E[i][j];
}

int main(){
    int m=3, n=3;
    int E[100][100] = {
        {1,3,1},
        {1,5,1},
        {4,2,1}
    };
    int res = minPath(m-1, n-1, E);
    printf("%d", res);
    return 0;
}

```

1. Rekürsif fonksiyona ait zaman karmaşıklığını, analiz için rekürans tree yapısı kullanarak, hesapla.
2. Bu problemi dinamik programlamayla çözen minPathDynamic rekürsif fonksiyonunu yaz.
3. minPathDynamic'e ait karmaşıklığı hesapla.
4. Problemi iteratif şekilde çözen minPath fonksiyonunu yaz.
5. Problemi iteratif şekilde çözen fonksiyonun toplam işlem sayısını toplam formülü kullanarak göster.