YZM 2116- VERİ YAPILARI

LAB#1: ALGORİTMA ANALİZİ

Lab Uygulaması 1

```
int subcalc1(int[] v1)
{
  int sum = 0;
  for (int i=0; i < v1.length; i++)
    sum = sum + v1[i]*v1[i]*v1[i];
  return sum;
}

int subcalc2(int[] v2)
{
  int sum = 0;
  for (int i=0; i < v2.length; i++)
    for (int j=0; j < i; j++)
      sum = sum + v2[i]*v2[j];
  return sum;
}

int calc(int[] v)
{
  return subcalc1(v) + subcalc2(v);
}
```

Yandaki kod bloğunda **calc()** fonksiyonun Big-O cinsinden karmaşıklığını hesaplayınız?

Lab Uygulaması 1

```
int subcalc1(int[] v1)
{
  int sum = 0;
  for (int i=0; i < v1.length; i++)
    sum = sum + v1[i]*v1[i]*v1[i];
  return sum;
}

int subcalc2(int[] v2)
{
  int sum = 0;
  for (int i=0; i < v2.length; i++)
    for (int j=0; j < i; j++)
      sum = sum + v2[i]*v2[j];
  return sum;
}

int calc(int[] v)
{
  return subcalc1(v) + subcalc2(v);
}
```

subcalc1()

$c1 + c3*N + c4$

subcalc2()

$c5 + c6*N^2 + c7$

calc()

$c1 + c3*N + c4 + c5 + c6*N^2 + c7$

$= N + N^2$

$= O(N^2)$

Lab Uygulaması 2

```
int power2(int n)
{
  int prod = 1;
  while (prod < n)
    prod = prod * 2;

  return prod;
}
```

Yandaki kod bloğunda **power2()** fonksiyonun Big-O cinsinden karmaşıklığını hesaplayınız?

Lab Uygulaması 2

```
int power2(int n)
{
  int prod = 1;
  while (prod < n)
    prod = prod * 2;

  return prod;
}
```

power2()
=c1+c2*n/2+c3
=O(logn)

Loop'un k kadar döndüğünü varsayarsak;
k adımında 2^i = n olur.
Her iki tarafın logaritmasını alırsak;
ilog2 = logn ve
i = log n olur.

Problemin **büyüklüğünü belli oranda(genelde ½) azaltmak** için **sabit bir zaman harcanıyorsa** bu algoritma O(log *N*)'dir.

Lab Uygulaması 3

- **Tanım:** Verilen bir tamsayı listesi içerisinde/dizisinde *elemanları* ***komşu olmak şartıyla*** hangi (bitişik) ***alt dizi*** en yüksek toplamı verir?

Örneğin:

- { -2, 11,-4, 13, -5, 2 }
- { 1, 2, -5, 4, 7, -2 }
- { 1, 5 ,-3, 4, -2, 1 }

Cevaplar:

- { -2, 11,-4, 13, -5, 2 } → Cevap = 20
- { 1, 2, -5, 4, 7, -2 } → Cevap = 11
- { 1, 5 ,-3, 4, -2, 1 } → Cevap = 7

Lab Uygulaması 3

Çözüm 1: Brute Force (Kaba Kuvvet) Algoritması

```
public int BruteForce(int[] a)
{
    int maxTop = 0;
    for (int i = 0; i < a.Length -1; i++)
       for (int j = i; j < a.Length - 1; j++)
       {
              int top = 0;
              for (int k = i; k <= j; k++)
              top += a[k];
              if (top > maxTop)
              {
                     maxTop = top;
                     int bas = i;  //alt dizinin başlangıcı
                     int son = j;  // alt dizinin sonu
              }
       }
        return maxTop;
}
```

Big O Analizi: $O(n^3)$

Lab Uygulaması 3

DAHA İYİSİ OLABİLİR Mİ?

Lab Uygulaması 3

Çözüm 2: Gelişmiş Brute Force (Kaba Kuvvet) Algoritması

```
public int BruteForceImproved(int[] a)
{
            int maxTop = 0;
            for (int i = 0; i < a.Length -1; i++)
            {
                int top = 0;
                for (int j = i; j <= a.Length -1; j++)
                {
                    top += a[j];
                    if (top > maxTop)
                    {
                        maxTop = top;
                        int bas = i;    // alt dizinin başlangıcı
                        int son = j;    // alt dizinin sonu
                    }
                }
            }
            return maxTop;
}
```

Big O Analizi: $O(n^2)$

Lab Uygulaması 3

DAHA İYİSİ OLABİLİR Mİ?

Lab Uygulaması 3

Çözüm 3: Doğrusal Algoritma

```
public int Dogrusal(int[] a)
{
            int maxTop = 0;
            int top = 0;
            for (int i = 0, j = 0; j <= a.Length - 1; j++)
            {
                top += a[j];
                if (top > maxTop)
                {
                    maxTop = top;
                    int bas = i;    // alt dizinin başlangıcı
                    int son = j;    // alt dizinin sonu
                }
                else if (top < 0)
                {
                    i = j + 1;
                    top = 0;
                }
            }
            return maxTop;
}
```

Big O Analizi: O(n)

Lab Uygulaması 3

SÜRE KARŞILAŞTIRMASI YAPALIM…

- Uygulamayı ders websayfasından indirerek test ediniz.