



YTÜ Skylab Kulübü  
Algorithm Games Challenge  
**Etap #3 ÇÖZÜMLERİ**

MATEMATİK BÖLÜMÜ

### **Cevap #1 (4 Puan)**

$29^4+29^3+29^2+29=707281+24389+841+29=732.540$  olacaktır.Bunları yuva kabul edersek,bu yuvalarda tünemesi gereken güvercin sayısı en az birisinde 2 dir.Dolayısıyla en az bir kelime 2 kez tekrarlanacaktır.

Güvercin yuvasıyla ilgili derlenmiş yazıyı çözümlerin devamında bulabilirsiniz..

---

### **Cevap #2 (3 Puan)**

3

pi sayısının sıralı rakamlarının 2 katının 1 fazlasının mod10 a göre kalanını yani birler basamağı sırayla yazılmıştır:

$$3 \rightarrow 3*2+1=7(\text{mod}10)$$

$$1 \rightarrow 1*2+1=3(\text{mod}10)$$

$$4 \rightarrow 4*2+1=9(\text{mod}10)$$

$$1 \rightarrow 1*2+1=3(\text{mod}10)$$

$$5 \rightarrow 5*2+1=1(\text{mod}10)$$

$$9 \rightarrow 9*2+1=9(\text{mod}10)$$

....

3.141592653...

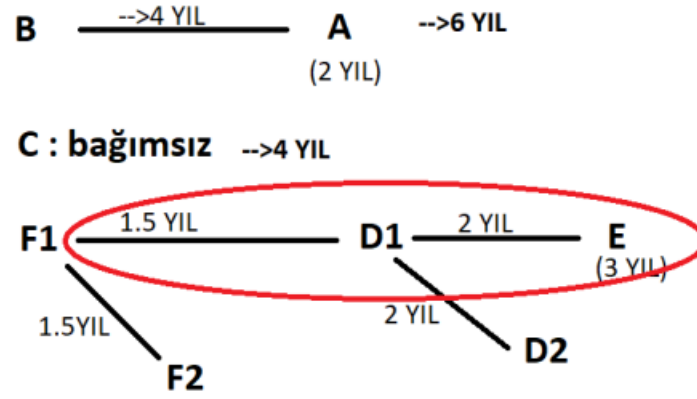
7.393195317

Aynı kural sorulan ifadeye uygulanırsa verilen cevaba ulaşılır

NOT: Verilen sayı dizisine uyan sistemli başka çözümler mevcut olabilir ve çözümlerinizden farklı cevaplar elde edilebilir.Sistemli çözümler paylaşılan çözümden bağımsız olarak değerlendirilecektir

---

**Cevap #3 (3 Puan)**



F1->D1->E=>6.5 YIL. Diğer bağlantılar en uzun bağlantı bitene kadar aynı süreçte tamamlanır.

## ALGORİTMA BÖLÜMÜ

### Cevap #4 (4 Puan):

```
#include <stdio.h>
#include <stdlib.h>

void fonksiyon(int* dizi,int diziuzunlugu);
int main(int argc, char *argv[]) {
    int diziuzunlugu;
    int dizi1[6]={3,6,4,5,2,1}, dizi2[6]={1,3,3,4,5,2},dizi3[6]={1,5,3,4,7,2},dizi4[6]={2,1,3,5,6,7};

    diziuzunlugu=6;
    printf("Dizi1 ciktisi: ");
    fonksiyon(&dizi1[0],diziuzunlugu);

    printf("Dizi2 ciktisi: ");
    fonksiyon(&dizi2[0],diziuzunlugu);

    printf("Dizi3 ciktisi: ");
    fonksiyon(&dizi3[0],diziuzunlugu);

    printf("Dizi4 ciktisi: ");
    fonksiyon(&dizi4[0],diziuzunlugu);
    return 0;
}

void fonksiyon(int* dizi,int diziuzunlugu){
    int i,j,indeks;
    for(i=0;i<diziuzunlugu;i++){
        indeks=0;
        for(j=0;j<diziuzunlugu;j++){
            if(*(dizi+j)==(i+1)){
                indeks++;
            }
        }
        if(indeks!=1){
            printf("0\n");
            break;
        }
        if(i==(diziuzunlugu-1)){
            printf("1\n");
        }
    }
}
```

## Cevap #5 (6 Puan)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

void kucuktenbuyugesiralamafonksiyonu(int dizi[], int elemanSayisi);

struct urunliste {
    char isim[30];
    int kilo;
};

int main(int argc, char *argv[]) {
    int fiyatetiketleridizisi[100];
    int m,n,i,j ,tmp[20];
    struct urunliste liste[10];
    char tmpisim[30];
    int minsum=0, maxsum=0, k=0;
    scanf("%d %d",&n,&m);

    srand(time(0));
    for(i=0;i<n;i++){
        fiyatetiketleridizisi[i]=(rand() % 9)+1;
    }
    kucuktenbuyugesiralamafonksiyonu(fiyatetiketleridizisi, n);

    for (i=0; i<m; i++) {
        scanf("%s",tmpisim);
        for (j=0; j<k &&(strcmp(tmpisim, liste[j].isim) != 0); j++) {
            ;;
        }
        if (j>=k) {
            strncpy(liste[j].isim, tmpisim, sizeof(tmpisim));
            liste[j].kilo=1;
            k++;
        } else {
            liste[j].kilo++;
        }
    }

    //her bir meyvenin kilosunu tutan dizi tmp dizisine aktarıldı küçükten büyüğe sıralandı
    for(i=0;i<k;i++){
        tmp[i]=liste[i].kilo;
    }
    kucuktenbuyugesiralamafonksiyonu(tmp, k);
    k++;
}
```

```

//her bir meyvenin kilosunu tutan dizi küçükten büyüğe sıralandı
//iki dizi de küçükten büyüğe sıralandığı için
//en büyük elemanı bulmanın yolu iki dizinin son elemanlarından geriye doğru birbiri ile çarpmak
//en küçük fiyatı bulmanın yolu iki dizinin ilk elemanlarından ileriye doğru birbiri ile çarpmak
//K ürün adeti bittiginde dizi sonlanacak.
    for(i=0;i<k;i++){
        maxsum+=tmp[k-1-i]*fiyatetiketleridizisi[n-1-i];
        minsum+=tmp[i]*fiyatetiketleridizisi[i];
    }
    printf("\nen büyük toplam=%d \n en küçük toplam=%d",maxsum,minsum);
    return 0;
}

void kucuktenbuyugesiralamafonksiyonu(int dizi[], int elemanSayisi)
{
    int temp;
    int i, j;
    for (i=1; i<elemanSayisi; i++)
    {
        for (j=0; j<elemanSayisi-i; j++)
        {
            if(dizi[j] > dizi[j+1])
            {
                temp = dizi [j];
                dizi [j] = dizi [j+1];
                dizi [j+1] = temp;
            }
        }
    }
}

```

## DERLENMİŞ YAZI..

***Benim kadar sürekli ve yoğun bir şekilde matematik üzerinde düşünen herkes, benim buluşlarımı ortaya koyabilir.( Johann Carl Friedrich Gauss )***



Güvercin yuvası ilkesi her ne kadar isminde güvercinlere yer verse de aslında bu güzel canlılarla uzaktan yakından bir alakası yoktur ancak temel prensibin kavranması verilen örneklerde hep bu canlıyla alakalı örneklemeler yapılmıştır.Peki nedir bu ilke?

Eğer  $n+1$  tane nesneyi(güvercini), $n$  tane kutu(güvercin yuvası) arasından seçersek, bu nesnelerden en az iki tanesi aynı kutudan gelmiş olacaktır. Benzer biçimde, herhangi bir ormandan seçilecek en az iki ağacın yaprak sayılarının aynı olmasını sağlayan yeterli ve gerekli şartlar nelerdir?Efsaneleşmiş Gauss hikayesinin araştırmasını size bırakıyoruz.

Güvercin Yuvası İlkesi sadece matematikte problem çözmede kullanılmıyor. Mesela bilgisayar biliminde çok fazla küme ve öge kullanıldığı için bunların birbiriyle çelişip çelişmediğini tespit etmek için güvercin yuvası ilkesi kullanılıyor (fakat daha güçlüsü). Aynı zaman da bilimde de kullanılmış oluyor. Mesela kuantum kuramı.Anlayacağınız aklınızın bir köşesinde güversin yuvası ilkesini bulundurmak sizin lehinize olacaktır..Ayrıntılı bilgi için internette bolca kaynak

bulabilirsiniz.