Manisa Celal Bayar Üniversitesi

Hasan Ferdi Turgutlu Teknoloji Fakültesi

Yazılım Mühendisliği

**Algoritma Analizi ve Tasarımı Dersi**

**Projesi**

**2017-2018**

Proje Ekibi

152802052 Taner YILDIRIM

152802042 Serkan KUMRU

2017 Manisa

## Projenin Amacı

N vezir problemin çözümüne yönelik olarak geliştirilmiş iki algoritmanın gerçekleştirimi yapılarak, bu algoritmaların etkinlikleri teorik ve pratik olarak (farklı girdi setleri için) değerlendirilmesidir. 8 Vezir bulmacası 8\*8 satranç tahtası üzerinde n vezirin birbirini yemeyecek şekilde konumlandırılmasıdır. ( Aynı satır, aynı sütun veya aynı çarpraz sırada iki veya fazla vezir bulunamaz.)

## Girdiler

N adet vezir.

## Beklenen Çıktılar

N vezirin birbirini yemecek konumda olacağı şekilde yerleştirilmesi.

## Kriterler

Vezirlerin birbirlerini yemeyecek şekilde konumlandırılması.

## Hata Kriterler

Vezirlerin birbirini yemesi.

## Mantığı

Öncelikle n=8 diyelim örnek üzerinden gidelin öncelikle her satır ve sutuna bir veziri random şeklinde yerleştiriyoruz.Ve ayrıyeten aynı boyutlarda oluşturduğumuz başka bir matristede her karenin kaç vezirin kapsama alanında olduğunu tutmak için kullanıyoruz.Önce ilk satırdaki vezir bulunur ve bulunduğu satırdaki en küçük değerli yere taşınır ve karelerin kapsaması dizisi yeniden değerlendirilir yani güncellenir.Şimdi ise taşıdığımız vezirin satırında başka vezir varmı bakıyoruz varsa onu taşıyoruz.Onun gittiği satirda varsa onu taşıyoruz taki satır sutunda tek olana kadar.Tek olunca en baştaki döngüye tekrar dönüp bu sefer 2. satırda ki veziri bulup aynı işlemi yapıyoruz.

## Program Kodu

1. Algoritma için;

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

int sayi = 0; //cozum sayisi

int \*tahta; //her eleman bir sirayi gosterir

int VEZIRLER; // vezir sayisi

int tahtaTamam(int sira){

int r;

for (r = 0; r < sira; r++)

if (tahta[sira] == tahta[r] || tahta[sira] == tahta[r] << sira-r || tahta[sira] == tahta[r] >> sira-r)

return 0;

return 1;

}

void EkranaBas(){// cozumu goster

int t, r;

printf("\n Cozum %d\n", ++sayi);

for (r = 0; r < VEZIRLER; r++) {

for (t = 1<<VEZIRLER-1; t > 0; t >>= 1)

printf(" %c", tahta[r] == t ? 'V' : 'O');

printf("\n");

}

}

void Diz(int sira) // bir sonraki siraya dizer

{

if (sira == VEZIRLER) // tum siralar dolu ve kontrol edilmis

EkranaBas();

else

for (tahta[sira]=1; tahta[sira]<1<<VEZIRLER; tahta[sira]<<=1)

if (tahtaTamam(sira))

Diz(sira+1); // bir sonraki sirayi dene

}

int main ()

{

printf("Vezir Sayisini giriniz :");

scanf("%d",&VEZIRLER);

tahta=(int\*) malloc(VEZIRLER\*sizeof(int)); //malloc ile vezir sayisina gore yeni dizi olusturuldu.

Diz(0); // ilk siradan basla

printf("\n\n%d vezir probleminin %u cozumu vardir.\n",VEZIRLER, sayi);

free(tahta);

return 0;

}

2.Algoritma için

#include<stdio.h>

#include<stdlib.h>

void Yazdir(int n,int x[]){

int i,j;

char c[100][100];

for(i=1;i<=n;i++) {

for(j=1;j<=n;j++) {

c[i][j]='O';

}

}

for(i=1;i<=n;i++) {

c[i][x[i]] ='V';

}

for(i=1;i<=n;i++){

for(j=1;j<=n;j++) {

printf(" %c",c[i][j]);

}

printf("\n");

}

}

int Kontrol(int x[],int n){

int i;

for(i=1;i<n;i++) {

if((x[i] == x[n]) || (i-x[i] == n-x[n]) || (i+x[i]==n +x[n])) {

return 0;

}

}

return 1;

}

void nVezir(int n){

int x[100],sayac=0,k=1,i;

x[k] = 0;

while(k != 0)

{

x[k] +=1;

while((x[k] <= n) && ( !Kontrol(x,k)))

{

x[k] +=1;

}

if(x[k] <= n)

{

if(k == n)

{

sayac++;

printf("\n Cozum %d\n",sayac);

Yazdir(n,x);

printf("\n",sayac);

}

else

{

k++;

x[k]=0;

}

}

else

{

k--;

}

}

printf("Toplam Cozum sayisi : %d",sayac);

}

int main()

{

int n;

printf("Vezir Sayisini giriniz :");

scanf("%d",&n);

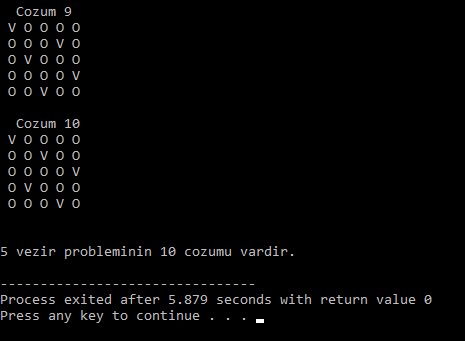
nVezir(n);

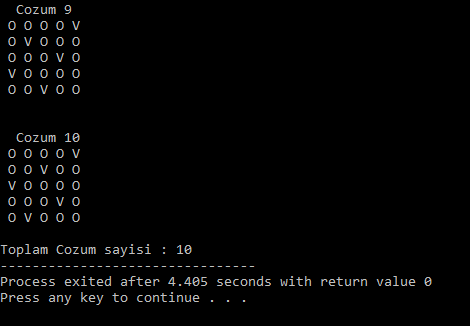
return 0;

}

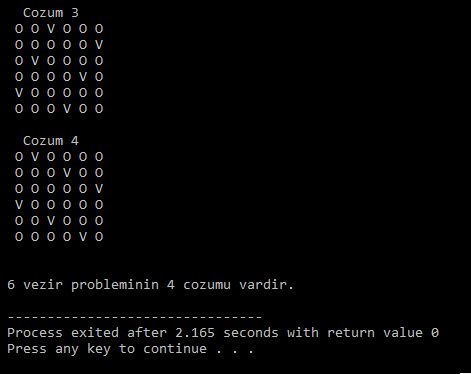
## Programların Ekran Çıktısı

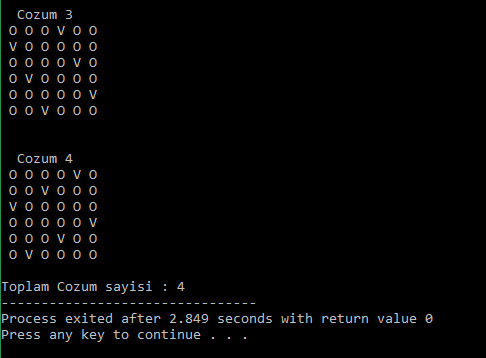
5 vezir için ekran çıktıları;



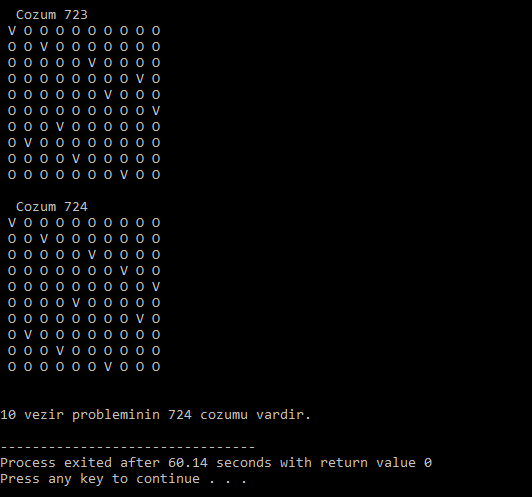


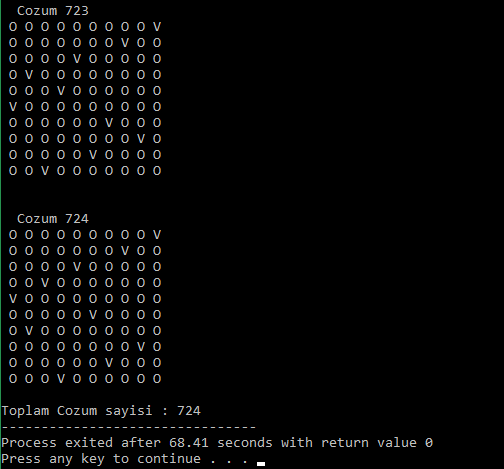
6 vezir için ekran çıktıları;





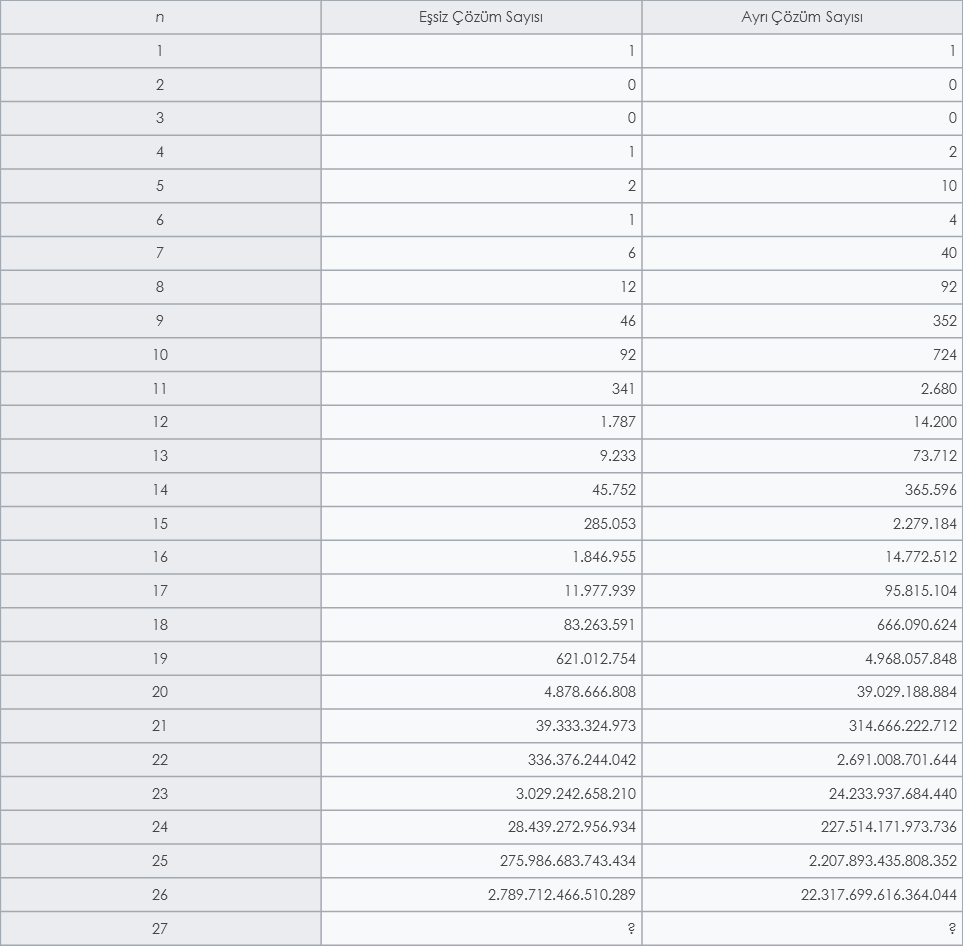
10 vezir için ekran çıktıları;





Not: 6×6'lık bir satranç tahtasında bulunan çözüm sayısının 5×5 boyutundaki bir satranç tahtasında bulunan çözüm sayısından az oluşu dikkat çekicidir.

***n vezir probleminin*** en son *n* = 26 değeri için çözümü bulunmuştur. Ancak, çok yüksek hesaplama gücüne gereksinim duyulduğundan *n* = 27 için çözüm henüz bulunamamıştır.



KAYNAKÇA

* <http://www.wikizero.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvU2VraXpfdmV6aXJfYnVsbWFjYXPEsQ>