**BLGM371 ALGORİTMALARIN ÇÖZÜMLENMESİ**

**ÇALIŞMA SORULARI 3**

1. Aşağıdaki özyinelemeli (rekürsiv) denklemleri istenilen yöntemle çözünüz.
2. T(n)=2\*T(n/2)+ n.lg n Iterasyon metodu ve master metodu
3. T(n)=(4/3)\*T(n/2) + lg n Master metodu (lg (4/3) = 0.415)
4. T(n)=8\*T(n/2) + n4 Master metodu
5. T(n)=3\*T(n/2) + n2 Master metodu (lg 3 = 1.5)
6. T(n)=2T(n/3) + n Iterasyon metodu
7. T(n)=3T(n/4)+n2 Iterasyon metodu
8. T8n)=3T(n/3)+n3 Iterasyon metodu
9. Aşağıdaki özyinelemeli (rekürsiv) denklemleri istenilen yöntemle çözünüz.
10. T(n)=T(n/4) + n Yerine koyma metodu ile O(n) olduğunu gösteriniz
11. T(n)=T(n-2) +n Yerine koyma metodu ile ϴ(n2) olduğunu gösteriniz
12. 7T(n/7) + n Yerine koyma metodu ile O(n.log7 n) olduğunu gösteriniz
13. T(n)=T(n/2) + 2n Yerine koyma metodu ile O(2n) olduğunu gösteriniz.
14. Aşağıdaki özyinelemeli (rekürsiv) denklemleri istenilen yöntemle çözünüz.
15. T(n)=2T(n-2)+2 Yerine koyma yöntemiyle T(n)= ϴ (2n) olup olmadığını test ediniz.
16. T(n)=4T(n/2) + n3 Master metodu
17. T(n)=2T(√n) + lg n Master metodu
18. T(n)=1, Eğer n=1

= Σ(i=2,...n) lg(i) Aksi halde Iterasyon yöntemiyle çözünüz.

1. T(n)=4 Eğer n=1

=2T(n/2)+4n Aksi halde Iterasyon yöntemiyle çözünüz.

1. T(n)=4T(n/16)+n0.3 Master metodu
2. T(n)=16T(n/4)+n Iterasyon metodu
3. Aşağıdaki özyinelemeli (rekürsiv) denklemleri çözünüz, çözümünüz asimptotik en yakın çözüm olsun.
4. T(√n)=T(√n)+n
5. T(n)=√n.T(√n)+n Yerine koyma yöntemiyle ϴ(n.lg lg n) olduğunu gösteriniz.
6. T(n)=T(n-1)+n2
7. T(n)=2T(n-1)+1
8. T(n)=T(99n/100)+T(n/100)+n
9. Aşağıdaki algoritmaların karmaşıklığını (T(n)) özyinelemeli denklemler olarak yazınız ve bu karmaşıklık denklemlerini uygun olan bir yönemle çözünüz.

|  |  |
| --- | --- |
| 1. *AlgoritmaA(n)*   R=0  For i=2 to n do  R=R+i\*AlgoritmaF(i)  AlgoritmaF(i)’nin hesaplama karmaşıklığı lg(i) dir. | 1. *AlgoritmaB(n)*   if n=1  return 1  else  T=AlgoritmaB(n/2)+AlgoritmaB(n/2)  for i=1 to n do  T=T+i  return T |
| 1. AlgoritmaC(n)   if n=1  return 1  else  T=AlgoritmaC(n/2)+AlgoritmaC(n/2)  for i=1 to n do  for j=1 to n do  T=T+i\*j  return T | 1. AlgoritmaD(n)   if n=1  return 1  else  R=0  for i = 2 to n do  R=R+AlgoritmaD(i-1)  return R |
| 1. AlgoritmaE(n)   İf n<=2  return 1  else  return (AlgoritmaE(|√n|+n) | 1. Algoritma Test(n,a)   İf n=1  return 1  else  for i=0 to n-1 do  a=a+1  Test(n/2)  Test(n/2)  return a |