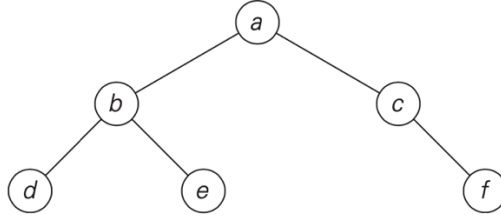


BLM2512 Veri Yapıları ve Algoritmalar Dersi – Final Öncesi Çalışma Soruları

1. Divide-and-Conquer yaklaşımını kullanarak N elemanlı bir dizideki maksimum elemanı bulan programı C dilinde yazınız.

2. Aşağıda verilen ağacı

- Preorder
- Inorder
- Postorder yaklaşımı ile hangi sırada gezeriz?



3. Aşağıdaki algoritmalardan kaç tanesi ön işlem gerektirir?

- Rabin-Karp Algoritması
- Naive String Matcher Algoritması
- Boyer-Moore Algoritması
- Boyer Moore Horspool Algoritması

- a. 1 b. 2 c. 3 d. 4 e. Hiçbiri

4. Boyer-Moore algoritmasının çalışma süresi y ekseninde, aranan kelimenin boyutu x ekseninde gösterilecek şekilde aralarındaki ilişkiyi grafik ile gösteriniz.
5. Verilen bir ağacın İkili Arama Ağacı(Binary Search Tree) özelliği taşıyıp taşımadığını kontrol eden ve sonucu ekrana yazdıran algoritmanın akış diyagramını çiziniz.
6. Bir ağacın dengeli olup olmadığını belirten kural nedir?
7. Grafın özel hali olan bir ağaçta düğüm ve kenar sayısı arasında nasıl bir ilişki mevcuttur?
8. Divide-and-Conquer yaklaşımını daha verimli hale getirmek için hangi donanımsal özelliğe sahip bilgisayarlar tercih edilebilir?

9. Verilen bir linkli listeyi, tersine çeviren algoritmayı tasarlayarak akış diyagramını çizin.

Ör:

12->23->34->45->56->NULL → 56->45->34->23->12->NULL

NULL → NULL

1->NULL → 1->NULL

10. Çift linkli ve sıralı bir linkli listede, yeni bir eleman eklemek için düğüm oluşturma ve link ekleme bağlantılarını doğru sırası ile çizerek gösteriniz.

Ör: 3↔5↔8↔15↔NULL → 3↔5↔8↔12↔15↔NULL

11. Push, pop, isEmpty, isFull, enqueue, dequeue fonksiyonlarının olduğunu varsayınız. Aşağıdaki kodlar çalıştığında hangi çıktıyı üretir?

a)

```
int main() {
    int i,j;
    int a[3] = {1,2,3};

    for (i=0;i<3;i++){
        push(a[i]);
        for (j=0;j<i;j++){
            push(a[j]);
        }
    }

    while(!isEmpty()) {
        int data = pop();
        printf("%d\n",data);
    }
    return 0;
}
```

b)

```
int main() {
    int i,j;
    int a[3] = {1,2,3};

    for (i=0;i<3;i++){
        enqueue(a[i]);
        for (j=0;j<i;j++){
            enqueue(a[j]);
        }
    }

    while(!isEmpty()) {
        int data = dequeue();
        printf("%d\n",data);
    }
    return 0;
}
```

12. Verilen iki stack'in eşit olup olmadığını kontrol eden algoritmayı tasarlayınız. Stack'lerdeki orijinal bilgileri kaybetmemeniz gereklidir.

Ör:

A	B	A	B	A	B	A	B
12	12	5	5			3	3
14	14	6	6			5	5
3	3	1	3			8	
8	8	4	4				
-5	-5						
"İki stack eşit"		"İki stack eşit değil"		"İki stack eşit"		"İki stack eşit değil"	

13. Elinizdeki bir queue veri yapısının içeriğini ters çevirmeniz isteniyor. İşlemi hangi veri yapısı ile gerçekleştirebilirsiniz? Kullandığınız fonksiyonların sırası ile birlikte anlatınız. Örnek için analizini yapınız.

Ör: 2 4 6 8 10 → 10 8 6 4 2

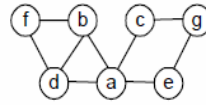
14. 4, 9, 3, 7, 2, 5, 8, 6 şeklinde verilen diziyi,
- Sırasıyla ekleyerek Min-Heap oluşturunuz. Her eleman ekleme adımındaki heap'i gösteriniz. 1 elemanını Min-Heap'e ekleyip son halini gösteriniz.
 - Sırasıyla ekleyerek Max-Heap oluşturunuz. Her eleman ekleme adımındaki heap'i gösteriniz. 9 elemanını Max-Heap'ten silip son halini gösteriniz.
15. N adet beyaz, N adet de siyahtan oluşan 2N adet diskiniz var. Bunlar, siyah-beyaz-siyah-beyaz-... şeklinde yanyana durmaktadır. Tüm siyah diskleri sona, tüm beyaz diskleri de başa taşımanız isteniyor. Yapabileceğiniz tek hareket, komşu olan iki diskin yerlerini değiştirmektir. Bu problemi çözebilmek için bir algoritma tasarlayıp akış diyagramını çiziniz. Karmaşıklığını analiz ederek yorumlayınız.



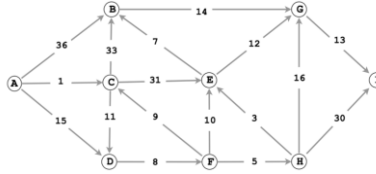
16. Aşağıdaki program parçasının **en iyi durum ve en kötü durum** çalışma zamanlarını hesaplayınız.

```
Algorithm Sample(A[0..n-1])
  for i = 1 to n-1 do
  {
    j = i-1
    while j ≥ 0 and A[j] > A[j+1] do
    {
      swap(A[j], A[j+1])
      j ← j-1
    }
  }
```

17. Aşağıda verilen grafta işleme a düğümünden başlayarak grafi **BFS ve DFS** yöntemleri ile gezdiğinizde **sırası ile hangi düğümlere uğranacağını** ayrı ayrı gösteriniz. (Düğümmler arasında ilerlerken işlemleri alfabadeki harf sırası ile yapınız. Örneğin a'dan önce b'ye sonra c'ye vb.)



18. Aşağıdaki grafta A düğümünden I düğümüne olan en kısa yolu Dijkstra En Kısa Yol algoritması ile bulunuz. Yolu ve yolun uzunluğunu veriniz.



19. Aşağıdaki dizinin böl ve yönet yaklaşımı ile Merge Sort ve QuickSort kullanılarak sıralanmasını ayrı ayrı adım adım gösteriniz.

8	3	2	9	7	1	5	4
---	---	---	---	---	---	---	---

20. Stable sort nedir? İki örnek veriniz.

21. Bir grafın kapalı bir çevrim(cycle) oluşturup oluşturmadığının **DFS yöntemi** ile bulan algoritmanın programını C dilinde yazınız. Aşağıdaki örnek üzerinde işlem adımlarını gösteriniz.



22. Verilen N elemanlı küçükten büyüğe sıralı bir dizide, $X \leq Y$ şartını sağlayan X ve Y değerleri için dizinin X'den büyük ve Y'den küçük sayılarının bulunduğu bölgenin başlangıç ve bitiş adreslerini bulan algoritmayı **böl ve yönet yaklaşımı** ile $O(\log N)$ karmaşıklıkla çözen algoritmanın akış diagramını çizip programını yazınız.

Örnek:

A[0..8] dizisi:

3	5	7	9	11	13	18	22	29
---	---	---	---	----	----	----	----	----

X = 8 ve Y = 24 değerleri için, başlangıç adresi 3, bitiş adresi 7'dir.