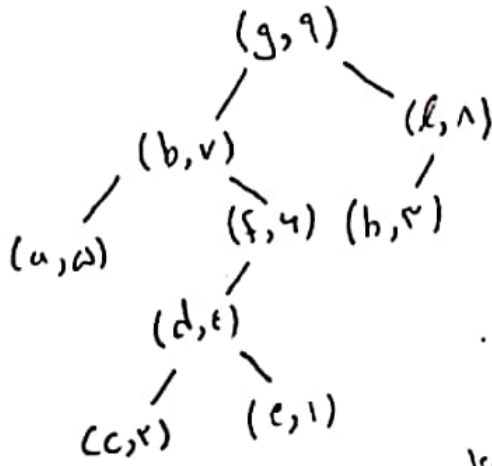


سؤال ۱: الف)



ب) ①  $l$  را برابر اندیس بزرگ‌ترین عضو  $prio$  قرار دهیم.

②  $(key[l], prio[l])$  را برابر  $root$  قرار دهیم.

③ الگوریتم را به ازای  $key[n+1], prio[n+1]$  و  $key[n+1], prio[n+1]$  اجرا کنیم.

در صورتی که  $root = root.right$ ، اجرای کنیم. در صورتی که  $(l+1 \neq n)$  باشد.

④ الگوریتم را به ازای  $key[l-1], prio[l-1]$  و  $key[l-1], prio[l-1]$  اجرا کنیم. در صورتی که  $(l-1 \neq 0)$  باشد.

سؤال ۲:

① \* برو به چپ‌ترین زیردرخت

\* چاپ کن

\* برو به راست، اگر نبود برو به چپ ②

⑤ \* اگر فرزند چپ برود، برو بالا، در غیر این صورت برو به چپ ③

\* چاپ کن

\* اگر راست موجود بود، برو به راست، چپ ④

⑥ \* این قدر بالا برو تا زیر درخت چپ باشی (اگر بالاتر  $null$  بود، برنامه را تمام کن).

\* یکی دیگر برو بالا و برو به چپ ⑤

```
void problem2 (Node head){
```

```
    while (true){
```

①

```
        while (head.left != null) head = head.left;
```

```
        print (head.value);
```

```
        if (head.right != null) head = head.right;
```

```
    } else while (true){
```

②

```
        if (head.parent.left == head) head = head.parent;
```

```
        else {
```

```
            while (true){
```

③

```
                if (head.parent == null) return;
```

```
                if (head.parent.left == head) break;
```

```
                head = head.parent;
```

```
            }
```

```
            head = head.parent
```

```
        }
```

```
        print (head.value);
```

```
        if (head.right != null){
```

```
            head = head.right;
```

```
            break;
```

```
        }
```

```
    }
```

```
} // first while ?
```

```
} // function
```

سؤال ٣:

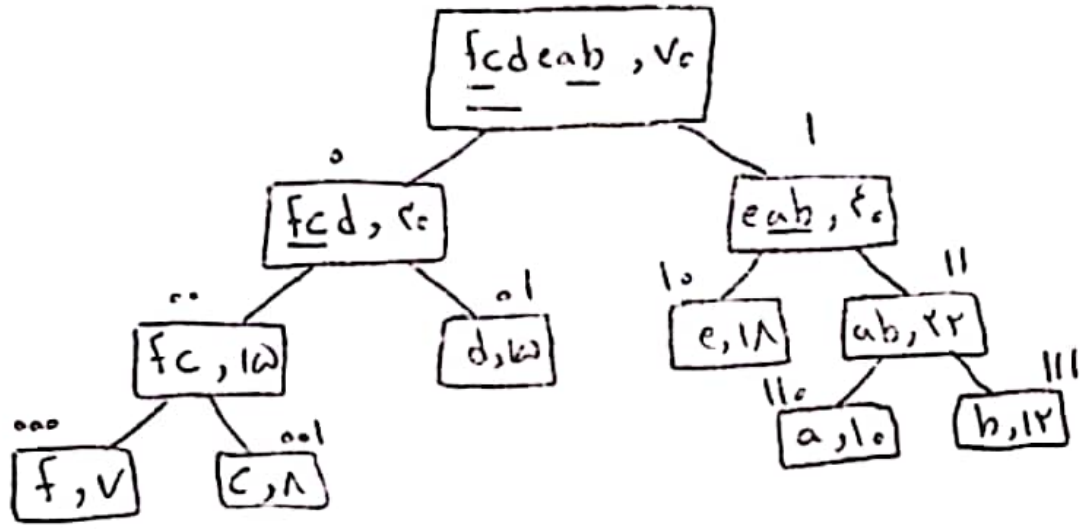
$e = 1\Lambda$   
 $d = 1\omega$   
 $b = 1\chi$   
 $a = 1\circ$   
 $c = 1\Lambda$   
 $f = V$

$\rightarrow$ 
 $e = 1\Lambda$   
 $d = 1\omega$   
 $f_c = 1\omega$   
 $b = 1\chi$   
 $a = 1\circ$

$\rightarrow$ 
 $ab = 2\chi$   
 $e = 1\Lambda$   
 $d = 1\omega$   
 $f_c = 1\omega$

$\rightarrow$ 
 $\underline{fcd} = 2\circ$   
 $ab = 2\chi$   
 $e = 1\Lambda$

$\rightarrow$ 
 $e \underline{ab} = 2\circ$   
 $\underline{fcd} = 2\circ$   
 $\rightarrow \underline{fcd} e \underline{ab} = V\circ$



$$V\circ (V\circ + 1\circ + 1\circ + 1\chi) + 2\circ (1\omega + 1\Lambda) = 2V\circ + 4\chi = 111 + 44 = 1VV$$

سؤال ٤:

$f\chi\chi\chi\omega 1\Lambda V$

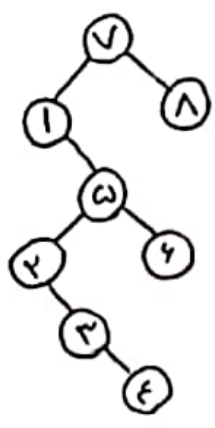
$\rightarrow$ 
 $f\chi\chi\chi\omega 1$

$\rightarrow$ 
 $f\chi\chi \chi \omega$

$\rightarrow$ 
 $f\chi\chi$

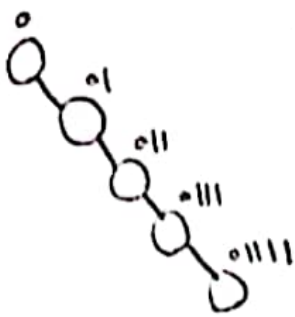
$\rightarrow$ 
 $f\chi$

$\rightarrow$ 
 $f$



سؤال ٥:

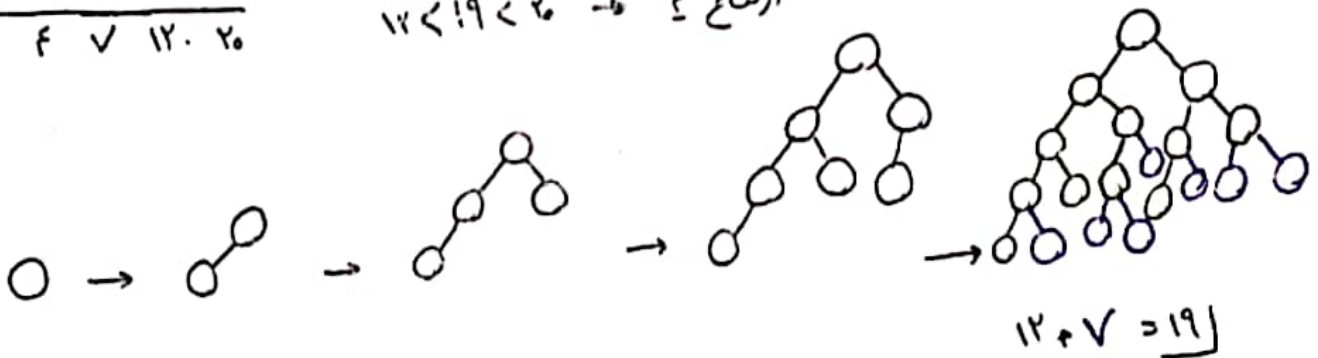
شکل درست این کدینگ - شکل درسته است و اگر انداز کدالترها زیاد باشه، طول کدها، خیلی بزرگ و بشود، در واقع این کدینگ، برای کدها، صرفاً برای درک یک تعداد کدالترها برای آن مهم است، در صورتی که کدینگ ها همین، به میزان تفاوت تعداد کدالترها نیز ترمیم کنند. ارجح تفاوت را زیاد و توان متوجه شده که تعداد کدالترها با تکرار تقریباً برابر داشته باشیم.



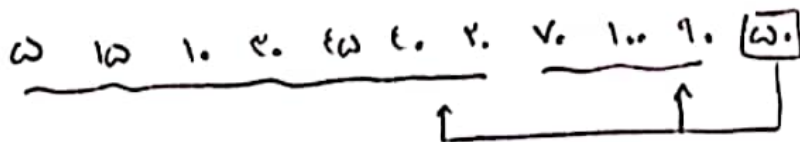
0	1	2	3	4	5
1	2	4	7	12	20

ارتفاع 4 → 12 < 19 < 20

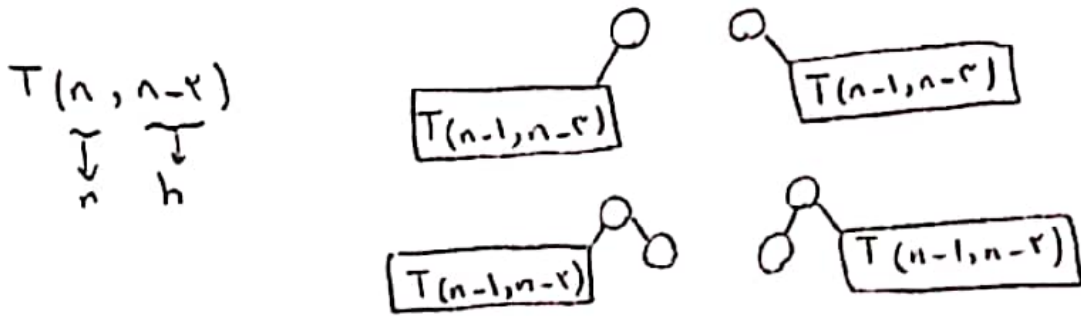
سؤال 2:



سؤال 3:



سؤال ۱:



$$T(n, n-2) = 2T(n-1, n-3) + 2T(n-2, n-4) = 2T(n-1, n-3) + 2^{n-2}$$

$$\rightarrow T'(n) = 2T'(n-1) + 2^{n-2} = 2T'(n-1) + \frac{2^n}{4}$$

{ عناصر: (n-2)  
ناعناصر: (n-2)

$$T'(n) = A2^n + Bn \cdot 2^n \rightarrow T(n, n-2) = A2^n + Bn \cdot 2^n$$

$$\left. \begin{array}{l} T(3, 1) = 1 \\ T(4, 2) = 4 \end{array} \right\} \rightarrow \left. \begin{array}{l} 8A + 24B = 1 \\ 16A + 48B = 4 \end{array} \right\} -$$

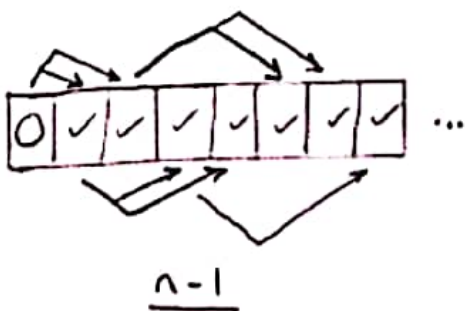
$$\rightarrow 16B = 4 \rightarrow B = \frac{1}{4} \rightarrow A = \frac{-5}{8} \rightarrow T(n, n-2) = -\frac{5}{8} \cdot 2^n + n \cdot 2^{n-2}$$

$$= 2^{n-3} (2n - 5)$$

سؤال ۱:  $O(k)$  چرا بیشتر نه: یویش میان ترتیب با  $O(n)$  انجام می‌شود، یویش را تا  $k$  امین عنصر ادامه می‌دهیم  
 کمتر نه: فرض کنید ارتفاع درخت  $n-1$  باشد، در این صورت حداقل کار، یویشی مسیر ریشه تا عنصر مورد نظر است.

سؤال ۱: حالت با  $i=0$  اگر  $(a[i] < a[i+1])$  برقرار بود،  $false$  را برگردان.  
 تا  $i = 2^{\log(n+3)}$  اگر  $(a[i] < a[i+2])$  برقرار بود،  $false$  را برگردان.

③ true را برگردان



سؤال ۱: ما برای تشخیص حیب بودن حداقلتر  $n-1$  مقایسه نیاز داریم.