

امتحان میان ترم درس ساختمان داده ها - دانشگاه صنعتی ارومیه - ۳ آذر ۱۳۹۵ - مدت ۱۲۰ دقیقه

۱. جواب رابطه های بازگشتی زیر را بدون استفاده از قضیه اصلی به دست آورید (۴ نمره).

a. $T(n) = T(2n/3) + \lg^2 n$,

$T(1) = 1$

$k = \lg_{1.5} n \leftarrow n = (1.5)^k$

$$\begin{aligned} T(n) &= T\left(\frac{n}{1.5}\right) + \lg^2 n = T\left(\frac{n}{1.5^2}\right) + \lg^2 \frac{n}{1.5} + \lg^2 n \\ &= T\left(\frac{n}{1.5^k}\right) + \lg^2 \frac{n}{1.5^{k-1}} + \dots + \lg^2 \frac{n}{1.5} + \lg^2 \frac{n}{(1.5)^1} \end{aligned}$$

$$T(n) = 1 + \lg^2 1.5 (1^2 + 2^2 + \dots + k^2)$$

$\lg^2 a^b = b^2 \lg^2 a$

$$= 1 + \lg^2 1.5 \left[\frac{k(k+1)(2k+1)}{6} \right]$$

$$= O(k^3) = O(\lg^3 n)$$

b. $T(n) = T(n/4) + T(3n/4) + n^2$, $T(1) = 1$

$T(n) \rightarrow n^2$

$$\begin{aligned} &T(n) \rightarrow T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) \rightarrow T\left(\frac{n}{4^2}\right) + T\left(\frac{3n}{4^2}\right) + T\left(\frac{3n}{4^2}\right) + T\left(\frac{3n}{4^2}\right) \rightarrow \dots \\ &\frac{n^2}{4^2} + \frac{3^2 n^2}{4^2} = n^2 \left(\frac{1+3^2}{4^2} \right) = n^2 \left(\frac{10}{16} \right) \\ &\frac{n^2}{4^4} + \frac{3^2 n^2}{4^4} + \frac{3^2 n^2}{4^4} + \frac{3^4 n^2}{4^4} = n^2 \left(\frac{1+3^2+3^2+3^4}{4^4} \right) = n^2 \left(\frac{10+9}{16} \right) \\ &= n^2 \left(\frac{1+3^2}{4^2} \right)^2 = n^2 \left(\frac{10}{16} \right)^2 \end{aligned}$$

$$T(n) \leq \sum_{i=0}^{\lg n} n^2 \left(\frac{10}{16} \right)^i$$

$n^2 \left(\frac{10}{16} \right)^3$

$n^2 \left(\frac{10}{16} \right)^{\lg n}$

$$O(n^2) = n^2 \left(\frac{1 - \left(\frac{10}{16} \right)^{\lg n + 1}}{1 - \frac{10}{16}} \right)$$

$$= O \left(n^2 \left(\frac{10}{16} \right)^{\lg n + 1} \right)$$

۲. دو لیست پیوندی p و q شامل اطلاعاتی از نوع یکسان می باشند (برای مثال اعداد صحیح) و هرکدام از آنها به صورت صعودی مرتب شده هستند. تابعی به صورت زیر بنویسید که لیست های p و q را به صورت مرتب شده ادغام نموده و در لیست پیوندی یکطرفه r قرار دهد (۳ نمره).

Merge(class node p, class node q, class node r)

Merge(class node p, class node q, class node r)

{

class node z;

if x == null & y == null

return;

while (x != null & y != null)

{

if r == null

{

r = allocate-node();

z = r;

else

{

next(z) = allocate-node();

z = next(z);

}

if (element(x) < element(y))

{

element(z) = element(x)

{

x = next(x);

else

{

element(z) = element(y);

y = next(y);

}

} // while


```
while (x != null)
```

```
{ next(Z) = allocate-node();
```

```
  Z = next(Z);
```

```
  element(Z) = element(X);
```

```
  X = next(X);
```

```
}
```

```
while (y != null)
```

```
{ next(Z) = allocate-node();
```

```
  Z = next Z;
```

```
  element(Z) = element(y);
```

```
  y = next(y);
```

```
}
```

```
next(Z) = null;
```

```
} // Merge
```

۳. تابعی بنویسید که ترتیب عناصر قرار گرفته در یک پشته S را به کمک دو پشته S1 و S2 معکوس نماید (۳ نمره).

ReverseStack (Stack S)

{

Stack S1, S2;

while !IsEmpty(S)

{

C = pop(S);

push(S1, C);

}

while !IsEmpty(S1)

{

C = pop(S1);

push(S2, C);

}

while !IsEmpty(S2)

{

C = pop(S2);

push(S, C);

}

}

۴. در صورتی که برای پیاده سازی دو پشته از یک آرایه مشترک استفاده شده باشد، که هر کدام از پشته ها از یک سمت آرایه شروع می شوند، تابع `IsFull` را به صورت زیر برای بررسی پر بودن پشته ها بنویسید. (no نشان دهنده شماره پشته می باشد و `top1` و `top2` اندیس عناصر بالای پشته ها را مشخص می کنند.) (۳ نمره)

`IsFull(Stack S, int no)`

{

`return top1 + 1 == top2 ;`

}

۵. ساختمان داده یک نود از لیست پیوندی یکطرفه را بنویسید و الگوریتمی برای درج یک نود در ابتدای لیست بنویسید. (۳)

نمره)

```
class Node{
```

```
    private Object element;
```

```
    private Node next;
```

```
    Node()
```

```
    { this (null, null);
```

```
    }
```

```
    public Node (Object e , Node n)
```

```
    {
```

```
        element = e;
```

```
        next = n;
```

```
    }
```

```
    void setElement (Object nE) { element = nE; }
```

```
    void setNext (Node nN) { next = nN; }
```

```
    Object getElement () { return element; }
```

```
    Node getNext () { return next; }
```

```
} // class Node
```

```
Insert-First (L, x)
```

```
{
```

```
    First (L) ← Allocate-Node (x, First (L)) ;
```

```
    size (L) ← size (L) + 1;
```

```
}
```


۶. در خصوص سری فیبوناچی به سوالات زیر پاسخ دهید. (۴ نمره)

الف) الگوریتمی بازگشتی برای حل این مساله بنویسید و تحلیل زمانی نمایید.

ب) الگوریتمی غیر بازگشتی برای حل این مساله بنویسید و تحلیل زمانی نمایید.

الف)

```
int fib(int n)
{
    if (n <= 2)
        return 1;
    else
        return (fib(n-1) + fib(n-2));
}
```

$$F(n) = F(n-1) + F(n-2)$$

$$x^2 - x - 1 = 0 \quad \left\{ \begin{array}{l} x_1 = \frac{1+\sqrt{5}}{2} \\ x_2 = \frac{1-\sqrt{5}}{2} \end{array} \right.$$

$$F_n = t_1 \left(\frac{1+\sqrt{5}}{2} \right)^n + t_2 \left(\frac{1-\sqrt{5}}{2} \right)^n$$

$$F_0 = 0 = t_1 + t_2$$

$$F_1 = 1 = t_1 \left(\frac{1+\sqrt{5}}{2} \right) + t_2 \left(\frac{1-\sqrt{5}}{2} \right) \quad \left\{ \begin{array}{l} t_1 = \frac{1}{\sqrt{5}} \\ t_2 = \frac{-1}{\sqrt{5}} \end{array} \right.$$

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right)$$

$$= O \left(\left(\frac{1+\sqrt{5}}{2} \right)^n \right)$$

```
int fib(int n)
```

```
{
```

```
    if (n <= 2)
```

```
        return 1;
```

```
    int f1 = 1;
```

```
    int f2 = 1;
```

```
    for (i = 3; i <= n; ++i)
```

```
    {
```

```
        f = f1 + f2;
```

```
        f1 = f2;
```

```
        f2 = f;
```

```
    } // for
```

```
    return f2;
```

```
}
```

$$T(n) = \sum_{i=3}^n 1 = n - 3 = O(n)$$