



به نام خداوند بخشنده مهربان

ساختمان داده - تمرین شماره یک

amin.adibnia@gmail.com

تاریخ تحویل ۴ اسفند ۹۷

۱. موارد زیر را اثبات کنید. [10]

الف) اگر $f(n) = O(g(n))$ آن گاه $f(n) = \Omega(f(n))$ است و بالعکس.

ب) اگر $f(n) = O(g(n))$ آن گاه $f(n)^2 = O(g(n)^2)$ است و بالعکس.

۲. باتوجه به کد زیر به موارد زیر پاسخ دهید: [15]

الف) اگر آرایه S به طول n باشد، پیچیدگی زمان اجرای کد در بدترین حالت را بنویسید.

ب) اگر درون آرایه S عناصر ۳ و ۸ و ۵ باشد، آن گاه مرتبه زمانی اجرای کد را برحسب C_i ها بدست آورید. (در

هنگام get کردن از ۳ شروع می کنیم)

```

s = list                                c۱
result = list                           c۲
while(s is not empty)                   c۳
    element = s.get                       c۴
    if(element > 3)                       c۵
        i = size of result                c۶
        while(i != 0)                     c۷
            if(result[i-1] > element)      c۸
                result[i] = result[i-1]   c۹
                i = i - 1                  c۱۰
            else                           c۱۱
                break                       c۱۲
        endwhile                          c۱۳
    result[i] = element                   c۱۴
endif                                    c۱۵
endwhile                                c۱۶

```

۳. توابع زیر را براساس سرعت رشد از صعودی به نزولی بنویسید. [10]

$\log n, n^3, 2^n, n^3 - n, \sqrt{n}, n^n, n \log n, \sqrt{\log n}, (\log n)!$

۴. در تکه کد زیر، loop invariant را مشخص کنید. [5]

```
result = 1;
for(i = 1; i < n; i++)
    result = result * i;
```

۵. زمان اجرای کدهای زیر را بدست آورید. (ذکر رابطه بازگشتی کافی است) [25]
(الف)

```
int bs(int arr[], int l, int h, int x) {
    if (h >= l) {
        int mid = l + (h - l) / 2;
        if (arr[mid] == x)
            return mid;
        if (arr[mid] > x)
            return bs(arr, l, mid - 1, x);
        return bs(arr, mid + 1, h, x);
    }
    return -1;
}
```

(ب)

```
int maxSubArr(int arr[], int l, int h) {
    if (l == h)
        return arr[l];
    int mid = (l + h)/2;
    return max(maxSubArr(arr, l, mid),
               maxSubArr(arr, mid+1, h),
               maxC(arr, l, mid, h)); // O(n) function
}
```

(پ)

```
int sumDigit(int number, int num_d) {
    if (number == 0)
        return 0;
    return (number%10 + sumDigit(number/10, num_d-1));
}
```

(ت)

```
void rev(Node* head, int& count, int n) {
    if (!head)
        return;
```

```

rev(head->next, count, n);
count = count + 1;
if (count <= n)
    cout << head->data;
}

```

ث) (امتیازی)

```

void f(int arr[], int l, int h) {
    if (l < h) {
        int r = rand(l, h); //random number between l and h
        f(arr, l, r - 1);
        f(arr, r + 1, h);
    }
}

```

۶. پیچیدگی روابط زیر را به روش دلخواه بدست آوردید. [25] ($T(1) = 1$)

- a) $T(n) = T(\frac{n}{3}) + n$
- b) $T(n) = T(\frac{n}{4}) + T(\frac{3n}{4}) + n$
- c) $T(n) = 2T(\frac{n}{4}) + n^2$
- d) $T(n) = \sqrt{n}T(\sqrt{n}) + n$
- e) $T(n) = T(n-2) + n$

۷. مرتبه زمانی کدهای زیر را مشخص کنید. [20]

الف)

```

for(i = 1; i < n; i++)
    for(j = n; j > i; j--)
        // O(1)

```

ب)

```

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

```

پ)

```

for(i = 1; i < n; i++)
    for(j = 1; j < n; j=j*2)
        for(k = 1; k < n; k++)
            // O(1)

```

ت) (امتیازی)

```
for(i = n; i > 1; i=i/2)
    for(j = 1; j < i; j++)
        // O(1)
```

شاد باشید.