Dayou Du
N13825075
dd2645@nyu.edu

Homework 3 - SA
Fundamental Algorithms

2017-10-05

# Ex 2.24

**Algorithm 1** Ex 2.24

**Input:** number of integers n
**Output:** print all the permutations for integers 1..n

```
procedure PermuteDriver(n; ;)
    Create empty array Arr[1..n]
    for i ∈ [1, n] do
        Arr[i] ← i
    end for
    Permute(n, Arr)
end procedure
procedure Permute(k; ; Arr[1..n])
    if k == 1 then
        print(Arr[1..n])
    else
        for i ∈ [1, k] do
            Swap(Arr[i], Arr[k])
            Permute(k − 1, Arr)
            Swap(Arr[i], Arr[k]
        end for
    end if
end procedure
```

I got this question CORRECT

# Ex 2.31

At beginning we call $DFSPath(T, 1)$

**Algorithm 2** Ex 2.31

**Input:** Current vertex $v$ to be visit, the index $idex$ that name $v$ should be written in to the array $Names$
**Output:** Print the path from root to $v$

```
1:  global n set to a suitably large value, the array Names[1..n]
2:  global the vertices in tree T, the lists child[v] for each v in T
3:  procedure DFSPath(v, idex; ;)
4:      if v is a leaf then
5:          Names[idex] ← v
6:          print(Names[1..idex])
7:      else
8:          Names[idex] ← v
9:          for each child c in child[v] do
10:             DFSPath(c, idex + 1)
11:         end for
12:     end if
13: end procedure
```

I got this question CORRECT. Notes : Line 5 and line 8 can be merged before line 4

## Ex 2.32

### a)

**Algorithm 3** Ex 2.32a

**Input:** $T$, root of the tree
**Output:** colored tree as required

```
1:  procedure ColorDriver(T; ;)
2:      for each child c in childT do
3:          Color(c)
4:      end for
5:  end procedure
6:  procedure Color(v; ;)
7:      if v is a leaf then
8:          v.color ← Green
9:      else
10:         v.color ← Green
11:         for each child c in child[v] do
12:             Color(c)
13:             if c.color == Green then
14:                 v.color ← Red
15:                 return
16:             end if
17:         end for
18:     end if
19: end procedure
```

I got this question CORRECT. Notes: Line 8 and line 10 can be merged before line 7.

### b)

**Algorithm 4** Ex 2.32b

**Input:** $T$, root of the tree
**Output:** colored(boolean) tree as required

```
1:  procedure BooleanColorDriver(T; ;)
2:      for each child c in childT do
3:          BooleanColor(c)
4:      end for
5:  end procedure
6:  procedure BooleanColor(v; ;)
7:      if v is a leaf then
8:          v.bool ← True
9:      else
10:         v.bool ← True
11:         for each child c in child[v] do
12:             BooleanColor(c)
13:             v.bool ← v.bool And NOT c.bool
14:         end for
15:     end if
16: end procedure
```

I got this question CORRECT. Notes: Line 8 and line 10 can be merged before line 7.

**c)**

**Algorithm 5** Ex 2.32c, two function version

**Input:** $T$, root of the tree
**Output:** colored(boolean) tree in A's perspective

```
1: procedure BeginGame(T; ;)
2:     for each child c in child[T] do
3:         AtoMove(c)
4:     end for
5: end procedure
6: procedure AtoMove(v; ;)
7:     if v is a leaf then
8:         v.bool ← True
9:     else
10:        v.bool ← True
11:        for each child c in child[v] do
12:            BtoMove(c)
13:            v.bool ← v.bool And c.bool
14:        end for
15:    end if
16: end procedure
17: procedure BtoMove(v; ;)
18:    if v is a leaf then
19:        v.bool ← False
20:    else
21:        v.bool ← False
22:        for each child c in child[v] do
23:            AtoMove(c)
24:            v.bool ← v.bool Or c.bool
25:        end for
26:    end if
27: end procedure
```

I got this question CORRECT. Notes: Line 8 and line 10 can be merged before line 7. Line 19 and line 21 can be merged before line 18.

# Ex 2.57

**Algorithm 6** Ex 2.57

**Input:** number of vertices $n$, child lists $Child[1..n]$
**Output:** Return the name of the Root

**function** $FindRoot(n, Child[1..n];;)$
    Create empty Marker Array $visited[1..n]$ initialize all elements to $False$
    **for** $i \in [1, n]$ **do**
        **for each** $c$ in $Child[i]$ **do**
            $visited[c] \leftarrow True$
        **end for**
    **end for**
    **for** $i \in [1, n]$ **do**
        **if** $visited[i] == False$ **then**
            **return** $i$
        **end if**
    **end for**
**end function**

I got this question CORRECT.

## Ex 3.2

### a)

$n > 2^{32}$
I got this question CORRECT.

### b)

$n > 32$
I got this question CORRECT.

### c)

$n > 2^8$
I got this question CORRECT.

## Ex 3.3

In increasing order:
$(\frac{1}{2})^n < n^{\frac{1}{log_2 n}} < 2 < 1001n^{10010001} < (log_2 n)^{log_2 n} = n^{log_2 log_2 n} < n^{log_2 n} < 10010n + 1.000000000001^n < 1.0000001^n < (log_2 n)^n$
I got this question CORRECT. Notes: We should use $<<$ instead of $<$

## Ex 3.4

### a

$$\begin{cases} S(1) = 1 \\ S(n) = S(n-1) + \Theta(1) \quad , n > 1 \end{cases} \tag{1}$$

I got this question CORRECT.

### b

$$\begin{cases} S(1) = 1 \\ S(n) = S(n-1) + \Theta(1) \quad , n > 1 \end{cases} \tag{2}$$

I got this question CORRECT.

### c

$$\begin{cases} S(1) = 1 \\ S(n) = 2S(n-1) + \Theta(1) \quad , n > 1 \end{cases} \tag{3}$$

I got this question CORRECT.

### d

$$\begin{cases} S(1) = 1 \\ S(2) = 1 \\ S(n) = S(n-1) + S(n-2) + \Theta(1) \quad , n > 2 \end{cases} \tag{4}$$

I got this question CORRECT.

## Ex 3.5

True: a, b1, b, c1, c, d, f1, f, g
False: e1, e
I got this question CORRECT.

## Ex 3.6

$f(n) = (lgn)^{lgn}$

As I check, I think I got this question CORRECT. However this answer didn't listed on solution.Here's how we prove it:
$log(n^k) = k \times logn$ (1)
$log((logn)^{logn}) = loglogn \times logn$ (2)
$log(c^n) = n \times logc$ (3)
Apparently, (1) << (2) << (3)

## Ex 3.13

### a)

$$\begin{cases} T(1) = 1 \\ T(n) = \frac{T(n)}{3} + \frac{T(n-1)}{3} + \frac{2T(n-1)}{3} + \Theta(1) = \frac{T(n)}{3} + T(n-1) + \Theta(1) \quad ,n > 1 \end{cases} \tag{5}$$

I got this question CORRECT.

### b)

$$\begin{cases} Y(1) = 1 \\ Y(n) = \frac{31Y(n)}{3} + \frac{41Y(n)+1}{3} + \frac{59Y(n-1)+26Y(n-1)+1}{3} = \frac{31Y(n)}{3} + 42Y(n-1) + \frac{2}{3} \quad ,n > 1 \end{cases} \tag{6}$$

I got this question CORRECT.
The subtle error of this problem is: The factor for $Y(n)$ on the right side of the equation is greater than 1, plus both $Y(n)$ and $Y(n-1)$ cannot be negative, thus $Y(n)$ have to be $+\infty$ to satisfy this equation.

### c)

$$\begin{cases} S(1) = 0 \\ S(n) = \frac{S(n)+1}{3} + \frac{S(n-1)+1}{3} + \frac{S(n-1)+1+S(n-1)+1}{3} = \frac{S(n)}{3} + S(n-1) + \frac{4}{3} \quad ,n > 1 \end{cases} \tag{7}$$

I got this question CORRECT.

## Ex 3.17

$$\begin{cases} Name(1) = 1 \\ Name(2n) = 2Name(n) - 1 \quad ,n \geq 1 \\ Name(2n+1) = 2Name(n) + 1 \quad ,n \geq 1 \end{cases} \tag{8}$$

I got this question CORRECT.

## Ex 3.37

### a)

$$U(99) = 99 + U(89.1) = 188.1 + U(80.19) = 268.29 + U(72.171) = 340.461 + U(64.9539) = 1314.7695 \quad (9)$$

$$Z(99) = 99 + Z(90) = 189 + Z(81) = 270 + Z(73) = 343 + Z(66) = 409 + Z(60) = 1309 \quad (10)$$

$$U(99) > Z(99) \quad (11)$$

I got this question CORRECT.

### b)

The reason that $U(99) > Z(99)$ is, for some x, we have:

$$U(x) = x + (\frac{9}{10})x + (\frac{9}{10})^2 x + ... + (\frac{9}{10})^{n-1}x + 15 \times (\frac{9}{10})^n x \quad (12)$$

$$Z(x) = x + \lceil(\frac{9}{10})x\rceil + \lceil\frac{9}{10}\lceil(\frac{9}{10})x\rceil\rceil + ... + \underbrace{\lceil(\frac{9}{10})...\lceil(\frac{9}{10})x\rceil...\rceil}_{n\ of\ \frac{9}{10}s} + 15 \times \underbrace{\lceil(\frac{9}{10})...\lceil(\frac{9}{10})x\rceil...\rceil}_{n+1\ of\ \frac{9}{10}s} \quad (13)$$

Where in equation (12), $(\frac{9}{10})^{n-1}x > 65$ and $(\frac{9}{10})^n x \leq 65$, and in equation (13), $\underbrace{\lceil(\frac{9}{10})...\lceil(\frac{9}{10})x\rceil...\rceil}_{n\ of\ \frac{9}{10}s} > 65$

For the first n items in $U(x)$ and $Z(x)$, they are approximately the same, although $Z(x)$ is a little greater(or equal) than $U(x)$ for each of these items.

For the last 1 item in $U(x)$ and last 2 items in $Z(x)$, there's a constant factor 15, thus the last item in $U(x)$ is larger than the sum of last two items in $Z(x)$. This is what happened in $U(99)$ and $Z(99)$.

Note: Apparently, $x$ cannot be too large since the first $n$ elements also slightly affect the sums of the equations. Actually the there are only 7 ranges that satisfy $U > Z$
$[80.00000, 80.24690]$
$[88.88890, 89.16322]$
$[98.44878, 99.07026]$
$[109.51730, 110.07806]$
$[121.72730, 122.30896]$
$[135.27650, 135.89884]$
$[150.64110, 150.99872]$
I got this question CORRECT. Notes : I didn't figure out how to proof that there are no more numbers(other than the ones in these 7 ranges) that satisfy $U > Z$. Just the program cannot find anymore. The intuition only give us that $x$ cannot be very large, however dealing with real numbers is much more tougher with $ceil$

### c)

The answer is no.
Under this condition, the first $n$ items in $U(x)$ and $Z(x)$ are not changed. For the sum of these items, $Z(x)$ is slightly larger(or equal, for some x) than $U(x)$.
The rest item in $U(x)$ is $C$. The rest items in $Z(x)$ is $\underbrace{\lceil(\frac{9}{10})...\lceil(\frac{9}{10})x\rceil...\rceil}_{n\ of\ \frac{9}{10}s} + C$. Apparently, the later one is larger.
Thus, under this condition, for all $x$, $U(x) \leq Z(x)$
I got this question CORRECT. Notes: Each item in the equations is corresponding to each level in the recursion tree.