FA HW 3

Xiaoyu Xue

2017-09-29

1 Ex 2.31

```
global n set to a suitably large value, the array Names[1..n];
global the vertices in tree T, the lists child[v] for each v in T;
procedure DFSPath(v, i);
   Names[i] ← v;
   if v is a leaf then
      print(Names[1..i]);
   else
      foreach vertex w in child[v] do
         DFSPath(w, i+1);
      endfor
   endif
end_DFSPath;
```

2 Ex 2.32

2.a

```
global the vertices in tree T, the lists child[v] for each v in T;
procedure ColorGameTree(v);
   v.color ← GREEN;
   foreach vertex w in child[v] do
      ColorGameTree(w);
      if w.color = GREEN then
         v.color ← RED;
      endif
   endfor
end_ColorGameTree;
```

Initial Procedure Call:

```
procedure StartGame();
   ColorGameTree(T.root);
end_StartGame;
```

2.b

```
global the vertices in tree T, the lists child[v] for each v in T;
procedure BoolGameTree(v);
   v.bool ← True;
   foreach vertex w in child[v] do
      BoolGameTree(w);
      if w.bool then
         v.bool ← False;
      endif
   endfor
```

```
end_BoolGameTree;
```

Initial Procedure Call:

```
procedure StartGame();
   BoolGameTree(T.root);
end_StartGame;
```

2.c

```
global the vertices in tree T, the lists child[v] for each v in T;
procedure AtoMove(v);
   v.bool ← True;
   foreach vertex w in child[v] do
      BtoMove(w);
      if NOTw.bool then
         v.bool ← False;
      endif
   endfor
end_AtoMove;

procedure BtoMove(v);
   v.bool ← False;
   foreach vertex w in child[v] do
      AtoMove(w);
      if w.bool then
         v.bool ← True;
      endif
   endfor
end_BtoMove;
```

Initial Procedure Call:

```
procedure StartGame();
   foreach vertex v in child[T.root] do
      AtoMove(v);
      BtoMove(v);
   endfor
end_StartGame;
```

3 Ex 2.24

```
global array Numbers[1...n] store n numbers procedure permDFS(index);
   if index>n then
      Print Numbers[1...n];
      return
   endif
   for i = index to n do
      swap(Numbers[i], Numbers[index]);
```

```
        permDFS(i + 1);
        swap(Numbers[i], Numbers[index]);
    endfor
end_permDFS;
```

Initial Procedure Call:

```
procedure PermALL();
    for i = 1 to n do
        Numbers[i] ← i;
    endfor
    permDFS(1);
end_PermALL;
```

4 Ex 2.57

```
global array Child[1...n] store child lists of each vertex;
global array Parent[1...n] store vertex's parent vertex;
function FindRoot();
    for int i = 1 to n do
        Parent[i] ← i;
    endfor
    for int i = 1 to n do
        foreach j in Child[i] do
            Parent[j] ← i;
        endfor
    endfor
    for int i = 1 to n do
        if Parent[i] = ithen
            returni;
        endif
    endfor
end_FindRoot;
```

5 Ex 3.2

5.a

$n < 2^{32}$

5.b

$n < 32$

5.c

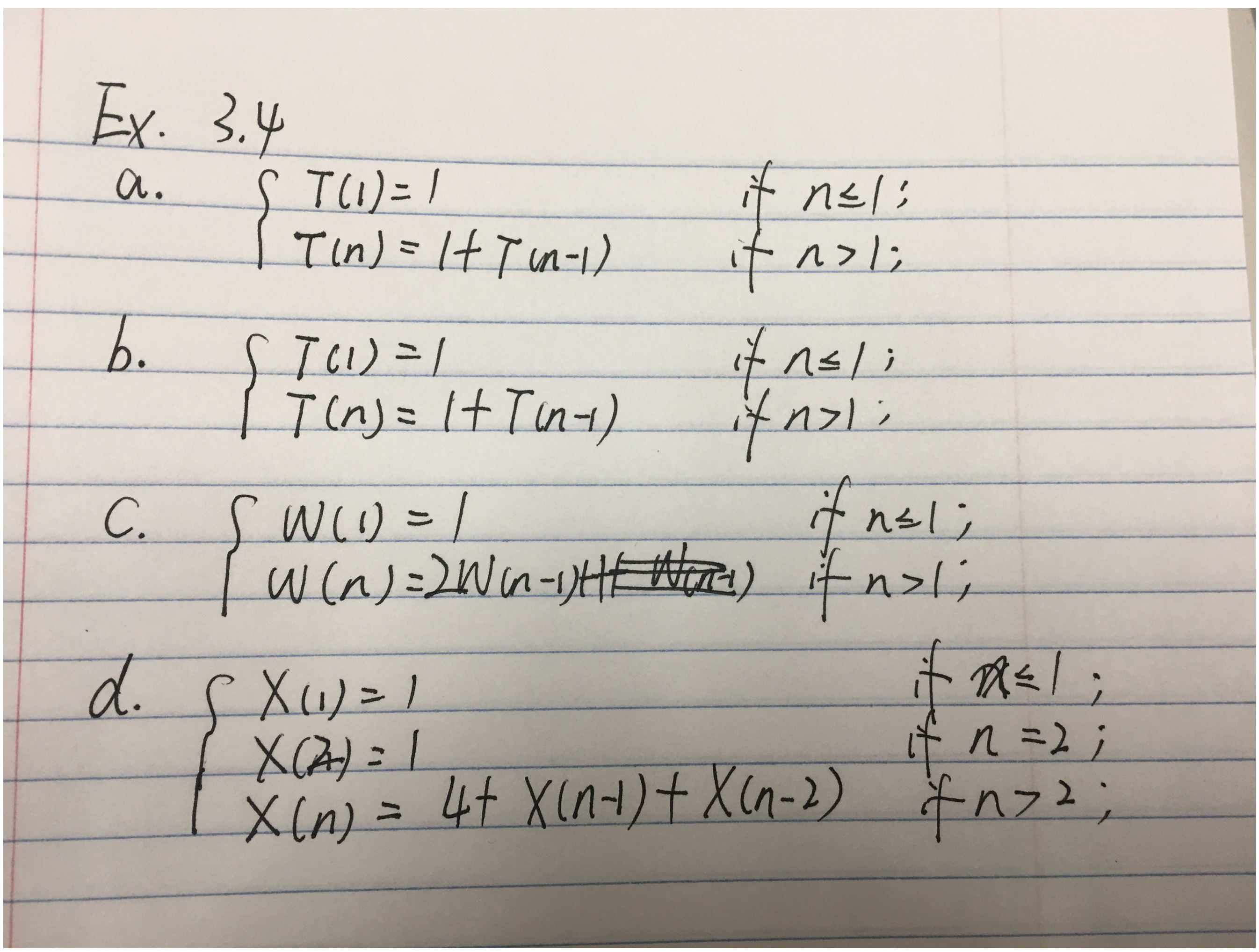
$n < 2^8$

6 Ex 3.3

$$(1/2)^n < n^{1/\log_2 n} = 2 < 1001n^{10010001} < n^{\log_2 \log_2 n} = (\log_2 n)^{\log_2 n} < n^{\log_2 n}$$

$$< 10010n + 1.000000000001^n < 1.00000001^n < (\log_2 n)^n$$

7 Ex 3.4



Ex. 3.4

a. $\begin{cases} T(1) = 1 & \text{if } n \le 1; \\ T(n) = 1 + T(n-1) & \text{if } n > 1; \end{cases}$

b. $\begin{cases} T(1) = 1 & \text{if } n \le 1; \\ T(n) = 1 + T(n-1) & \text{if } n > 1; \end{cases}$

c. $\begin{cases} W(1) = 1 & \text{if } n \le 1; \\ W(n) = 2W(n-1)+1 & \text{if } n > 1; \end{cases}$

d. $\begin{cases} X(1) = 1 & \text{if } n \le 1; \\ X(2) = 1 & \text{if } n = 2; \\ X(n) = 4 + X(n-1) + X(n-2) & \text{if } n > 2; \end{cases}$

Figure 1: *Exe 3.4*

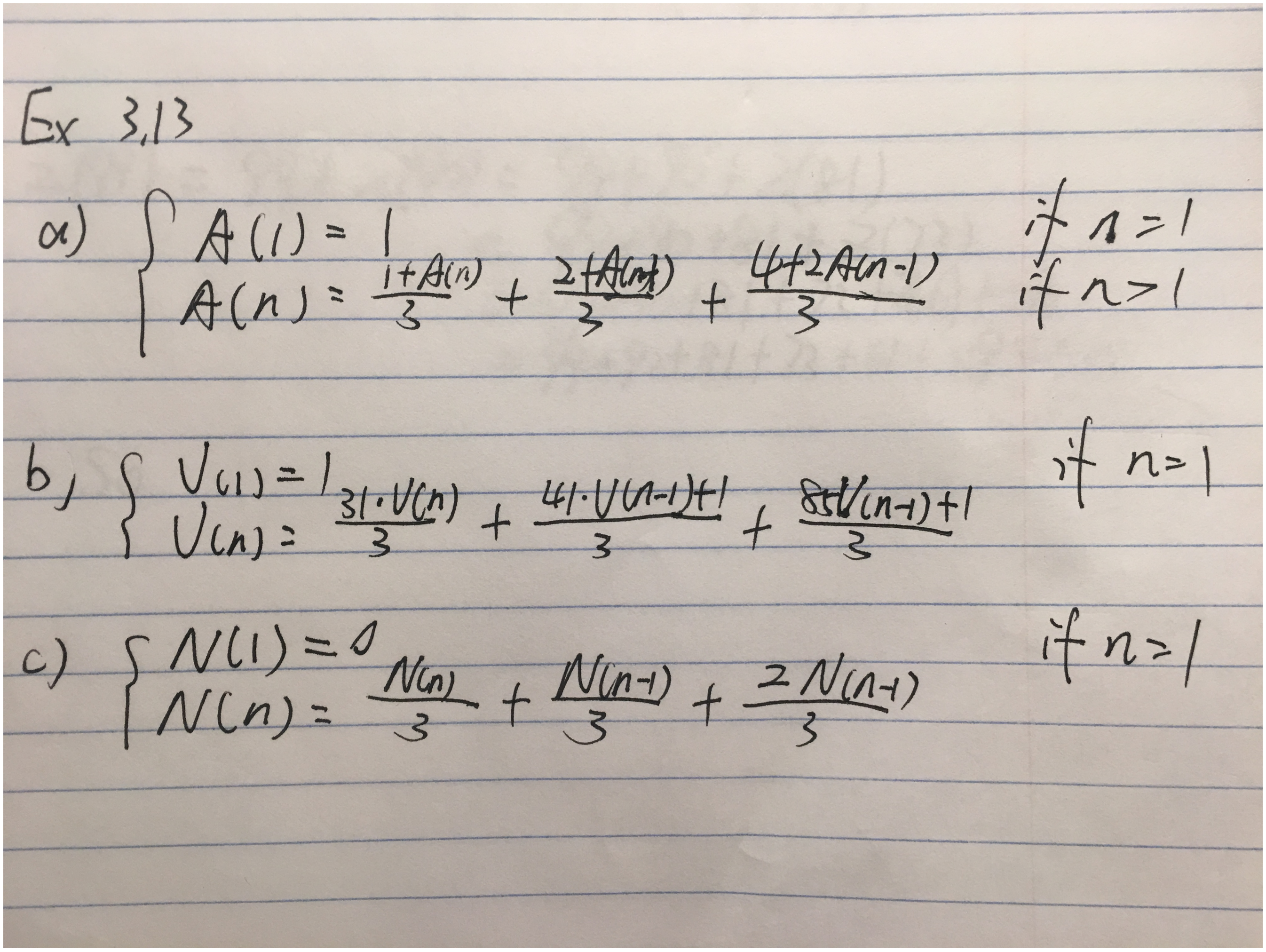
8 Ex 3.5

a). True
b1) True
b) True
c1) True
c) True
d) True
e1) False
e) False
f1) True
f) True
g) True

9 Ex 3.6

$f = n^{\log_2 n}$

10 3.13



Ex 3.13

a) $\begin{cases} A(1) = 1 & \text{if } n=1 \\ A(n) = \frac{1+A(n)}{3} + \frac{2+A(n-1)}{3} + \frac{4+2A(n-1)}{3} & \text{if } n>1 \end{cases}$

b) $\begin{cases} V(1) = 1 & \text{if } n=1 \\ V(n) = \frac{31\cdot V(n)}{3} + \frac{41\cdot V(n-1)+1}{3} + \frac{85V(n-1)+1}{3} \end{cases}$

c) $\begin{cases} N(1) = 0 & \text{if } n=1 \\ N(n) = \frac{N(n)}{3} + \frac{N(n-1)}{3} + \frac{2N(n-1)}{3} \end{cases}$

Figure 2: *Exe 3.13 Solution.*

11 Ex 3.17

$$\begin{cases} NAME(2n) = 2NAME(n) - 1 \\ NAME(2n+1) = 2NAME(n) + 1 \end{cases}$$

12 Ex3.37

12.a

U(99) = 99 + 89.1 + 80.19 + 72.171 + 64.9539 * 15 = 1314.7695
Z(99) = 99 + 90 + 81 + 73 + 66 + 59.4*15 = 1210
Hence, $U(99) > Z(99)$.

12.b

Although Z(x) grows faster than U(x), there is a special case when the equation switch from the second one to the first one. If the value of X is very close to 65 and the first equation multiple it by 15, it will make the gap bigger, which can cause the U bigger than Z.

12.c

No. Because there is not a multiplier that magnificent the gap between U and Z.