FA HW 4

Xiaoyu Xue

2017-10-07

1 Ex 2.12

1.a

```
global int sum is all pre-visited vertices' dat;
procedure DFS(v);
    sum ← sum + v.dat;
    v.ans ← sum;
    foreach vertex w in child[v] do
        DFS(w);
    endfor
end_DFS;
```

Initial Procedure Call:

```
procedure Driver();
    sum ← 0;
    DFS(T.root);
end_Driver;
```

1.b

```
global int sum is all pre-visited vertices' dat;
procedure DFS(v);
    foreach vertex w in child[v] do
        DFS(w);
    endfor
    sum ← sum + v.dat;
    v.ans ← sum;
end_DFS;
```

Initial Procedure Call:

```
procedure Driver();
    sum ← 0;
    DFS(T.root);
end_Driver;
```

1.c

```
global int sum is all pre-visited vertices' dat;
procedure DFS(v);
    DFS(v.left);
    sum ← sum + v.dat;
    v.ans ← sum;
    DFS(v.right);
end_DFS;
```

Initial Procedure Call:

```
procedure Driver();
   sum ← 0;
   DFS(T.root);
end_Driver;
```

1.d

```
global int sum is all have not yet been pre-visited vertices' dat;
procedure DFS(v);
   sum ← sum − v.dat;
   foreach vertex w in child[v] do
      DFS(w);
   endfor
   v.ans ← sum;
end_DFS;
```

```
procedure initDFS(v);
   sum ← sum + v.dat;
   foreach vertex w in child[v] do
      DFS(w);
   endfor
end_initDFS;
```

Initial Procedure Call:

```
procedure Driver();
   sum ← 0;
   initDFS(T.root);
   DFS(T.root);
end_Driver;
```

1.e

```
global int sum is all pre-visited vertices' dat;
procedure DFS(v);
   sum ← sum + v.dat;
   foreach vertex w in child[v] do
      DFS(w);
   endfor
   sum ← sum − v.dat;
   v.ans ← sum;
end_DFS;
```

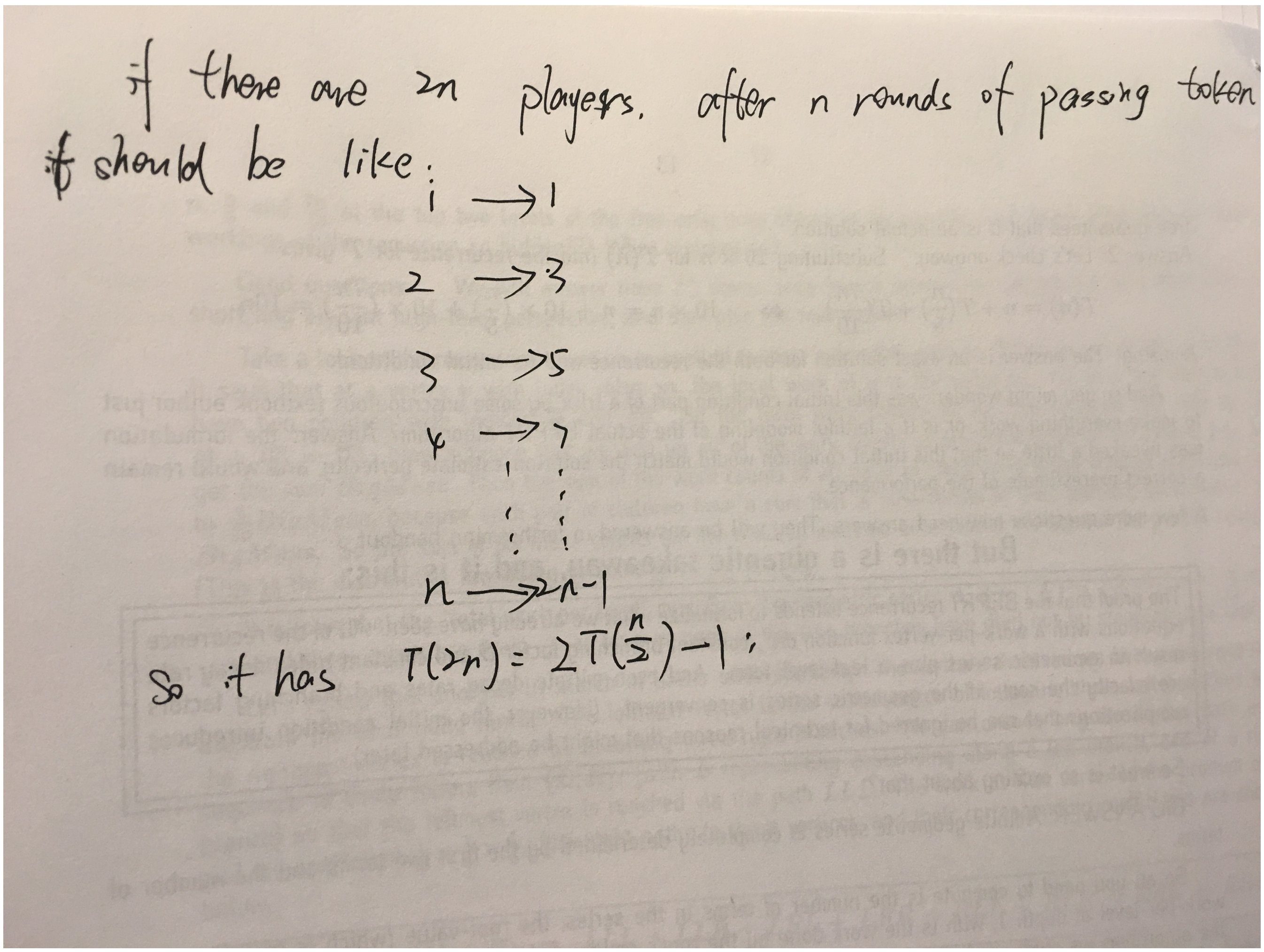
Initial Procedure Call:

```
procedure Driver();
   sum ← 0;
   DFS(T.root);
end_Driver;
```

1.f

Those vertices that have been preorder visited, but note postorder exited at the time that v is postorder exited, construct a path from the current vertex to the root of T.

2 Ex 3.17



if there are 2n players. after n rounds of passing token it should be like:

$1 \longrightarrow 1$

$2 \longrightarrow 3$

$3 \longrightarrow 5$

$4 \longrightarrow 7$

$\vdots \quad \vdots$

$n \longrightarrow 2n-1$

So it has $T(2n) = 2T(\frac{n}{2}) - 1$;

Figure 1: *Exe 3.17 Solution.*

$$
\begin{cases}
\mathrm{T}(1) = 1 & if\ n = 1; \\
\mathrm{T}(n) = 2\mathrm{T}(n/2) - 1 & if\ n\ is\ even; \\
\mathrm{T}(n) = 2\mathrm{T}((n-1)/2) + 1 & if\ n\ is\ odd;
\end{cases}
$$

3 Ex 3.18

3.a

The special value should separate the equation , so it has $T(n) = 1024n + 2T(n/2)$ and $T(n/2) = 4(n/2)^2$ and $T(n) = 4n^2$.

$$1024n + 4(n/2)^2 = 4n^2 \tag{1}$$

Hence, $n = 512$.

3.b

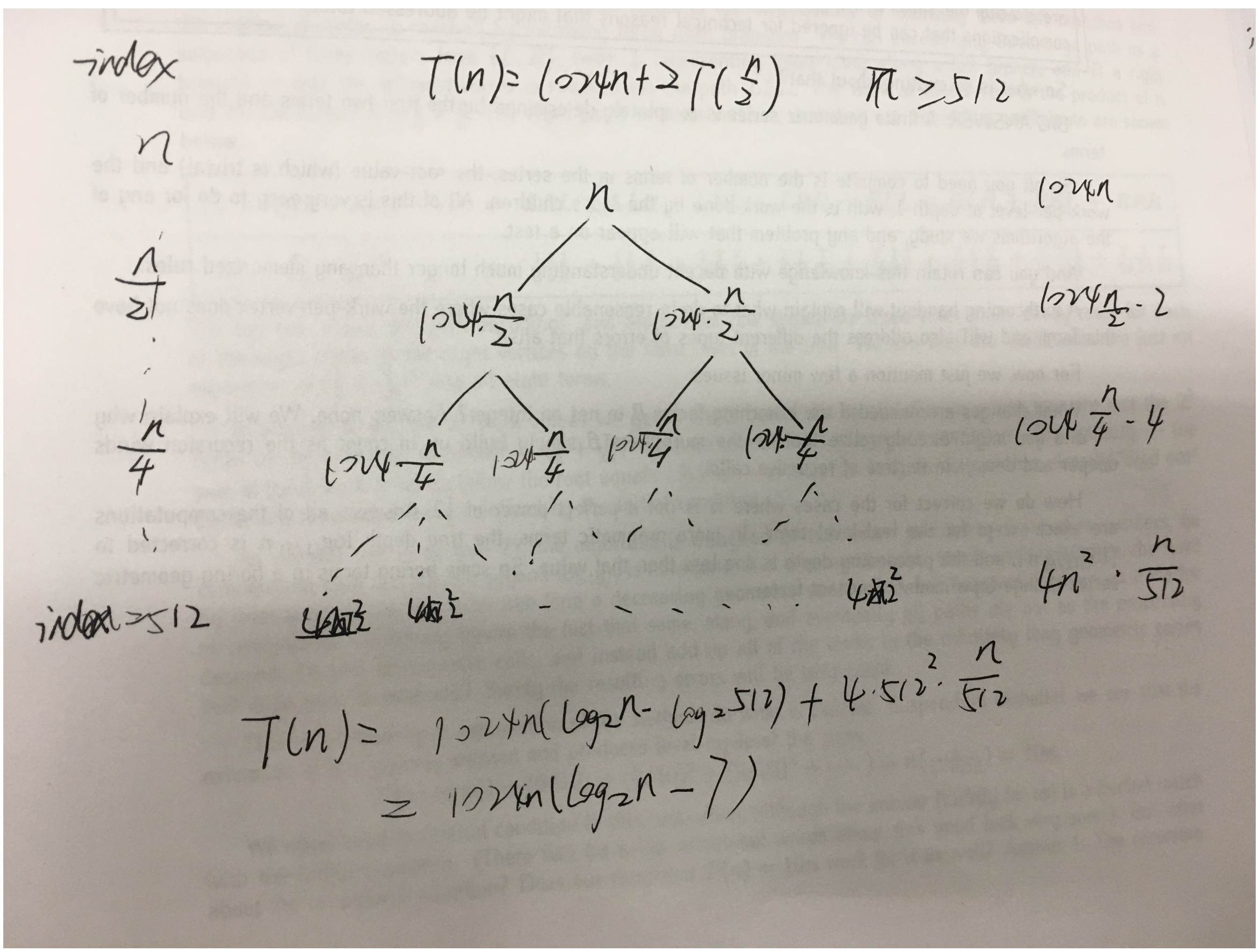


Figure 2: *Exe 3.18 Solution.*

So $\quad T(n) = 1024n(\log_2 n - 7)$.

4 Ex 3.19

4.a

$$\begin{cases} \mathrm{L(0)} = 2 & if\ k\ =\ 0; \\ \mathrm{L(k)} = 2 +\ 2\mathrm{L(k\text{-}1)} & if\ k\ >\ 0; \end{cases}$$

4.b

$$\begin{cases} \mathrm{N(0)} = 1 & if\ k\ =\ 0; \\ \mathrm{N(k)} = 3 +\ 4\mathrm{N(k\text{-}1)} & if\ k\ >\ 0; \end{cases}$$

4.c

$L(k) = 2^k + 2$ if k $>$ 0;
$N(k) = 3 + 4^k$ if k $>$ 0;
$S(k) = N(k) + 1 + 2L(k)$

5 Ex 3.28

5.a

$A(n) = n\log_2 n + n$

5.b

$B(n) = n\log_3 n + n$

5.c

$C(n) = n^2\log_2 n + n^2$

5.d

$D(n) = n^2\log_3 n + n^2$

5.e

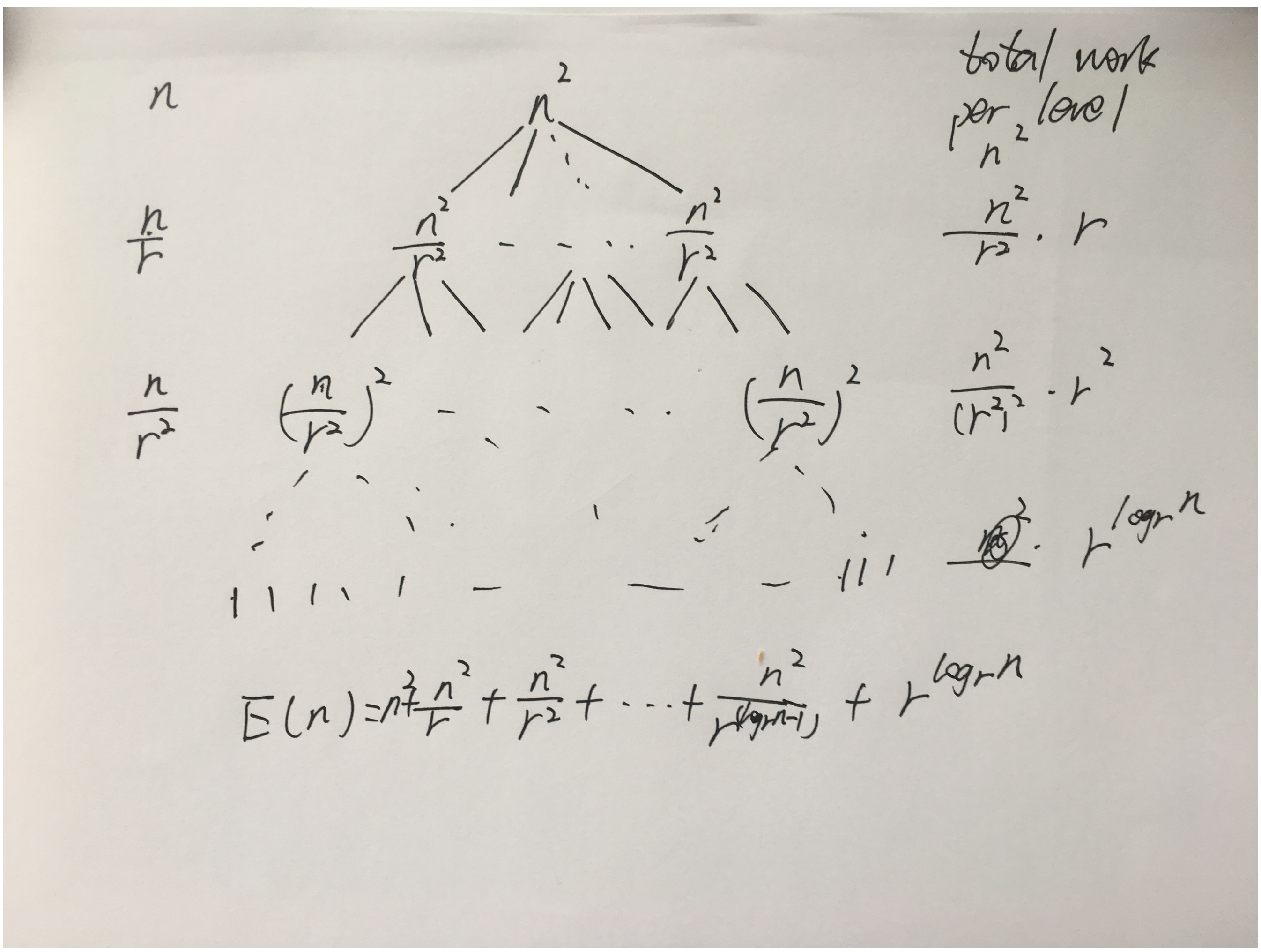


Figure 3: *Exe 3.28e Solution.*

5.f

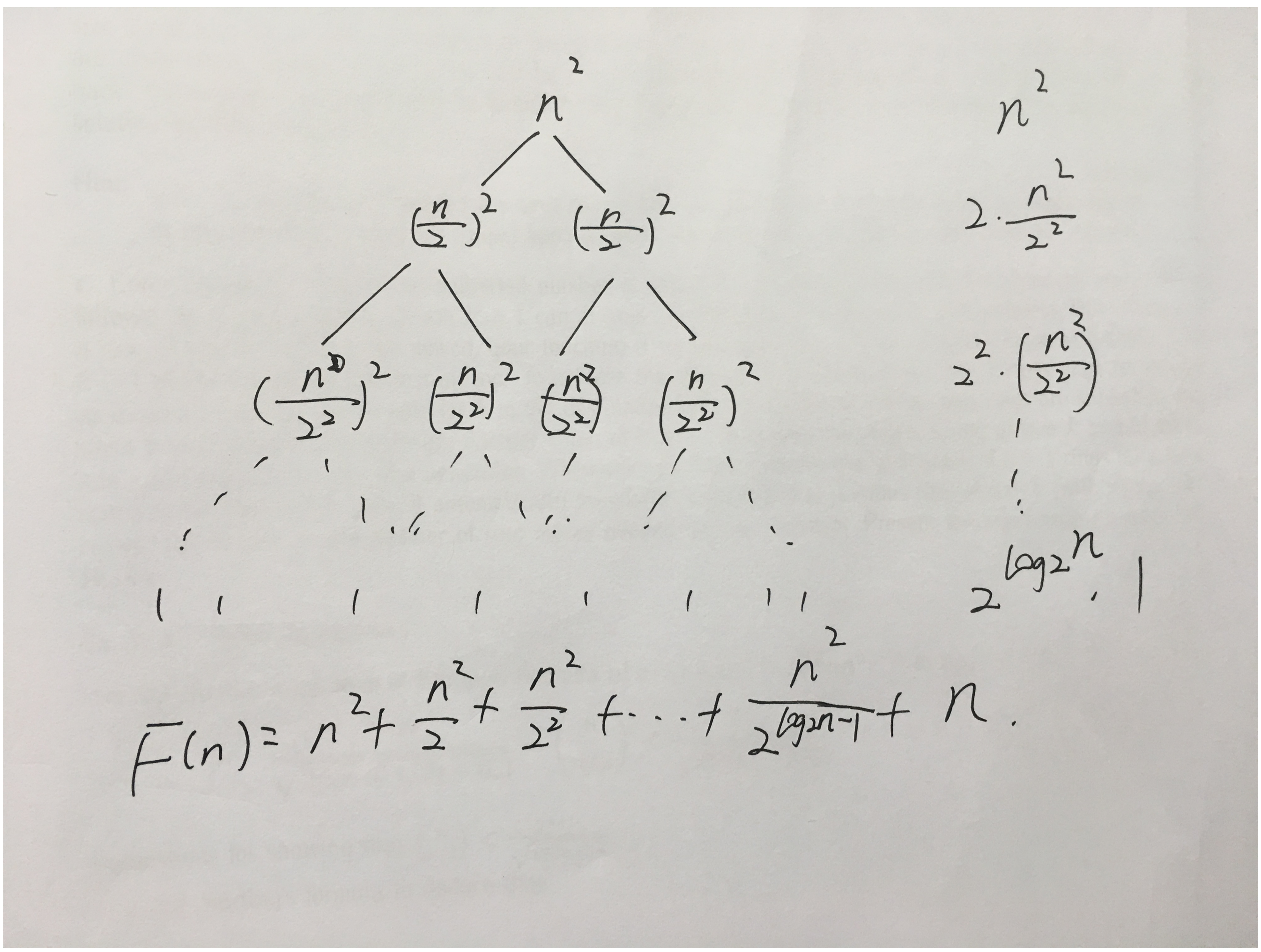


Figure 4: *Exe 3.28f Solution.*

5.g

5.h

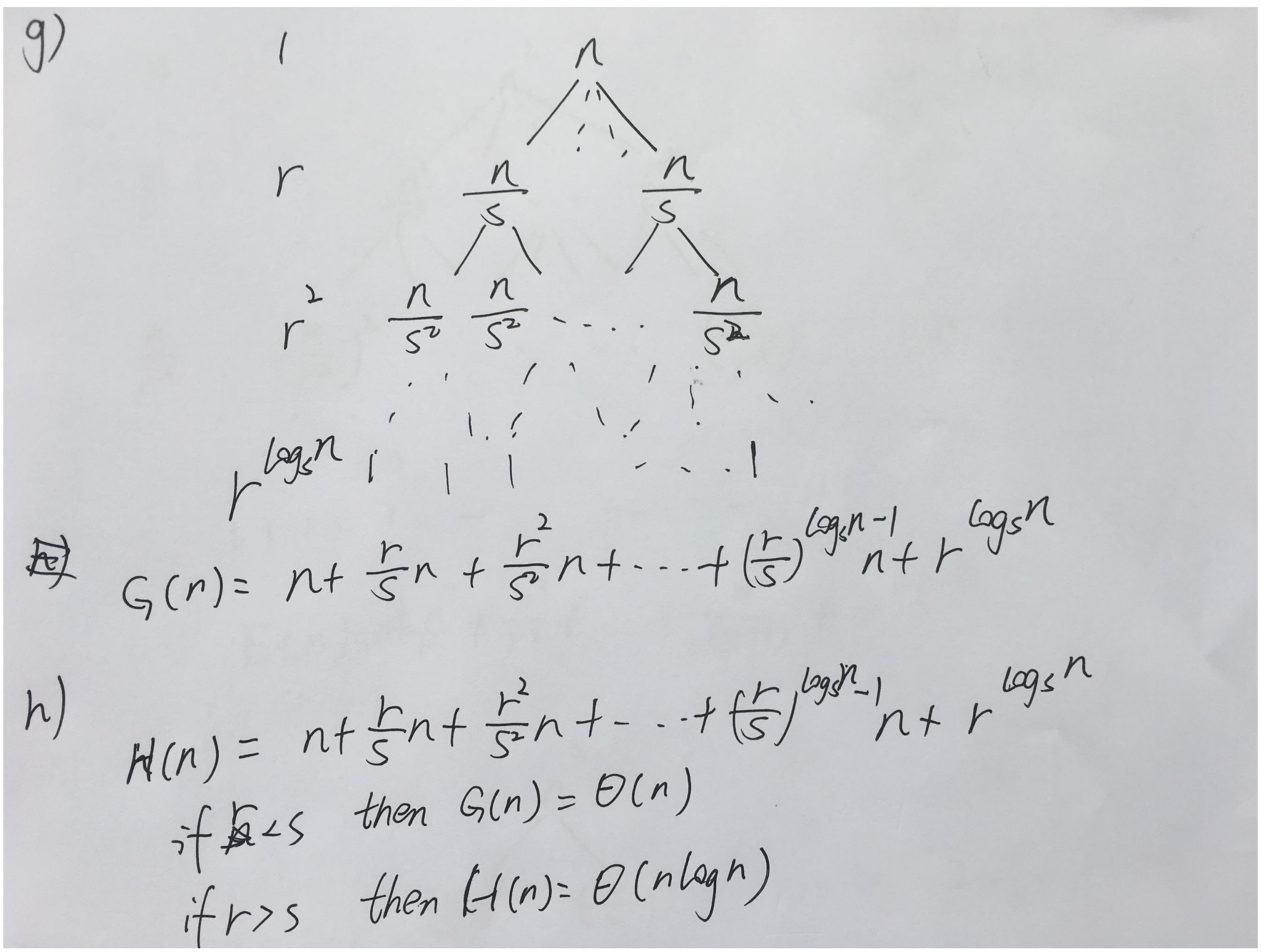


Figure 5: *Exe 3.28gh Solution.*

5.i

5.j

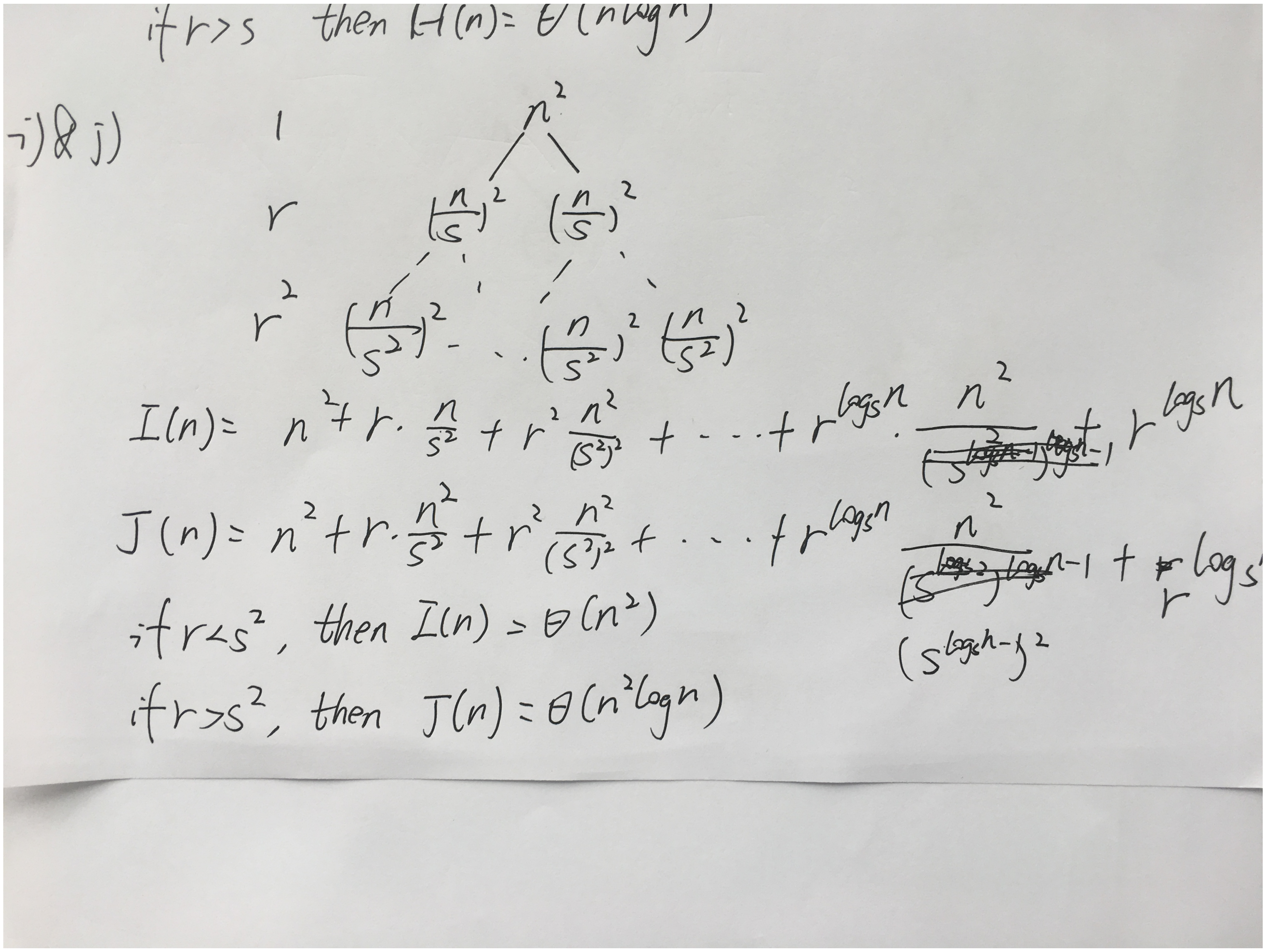


Figure 6: *Exe 3.28ij Solution.*

5.k

$K(n) = n^2(\log_3 n - 4) + 81 * 9^{(\log_3 n - 4}$

5.l

$L(n) = \log_9 n\sqrt{n} + \sqrt{n}$

6 3.general

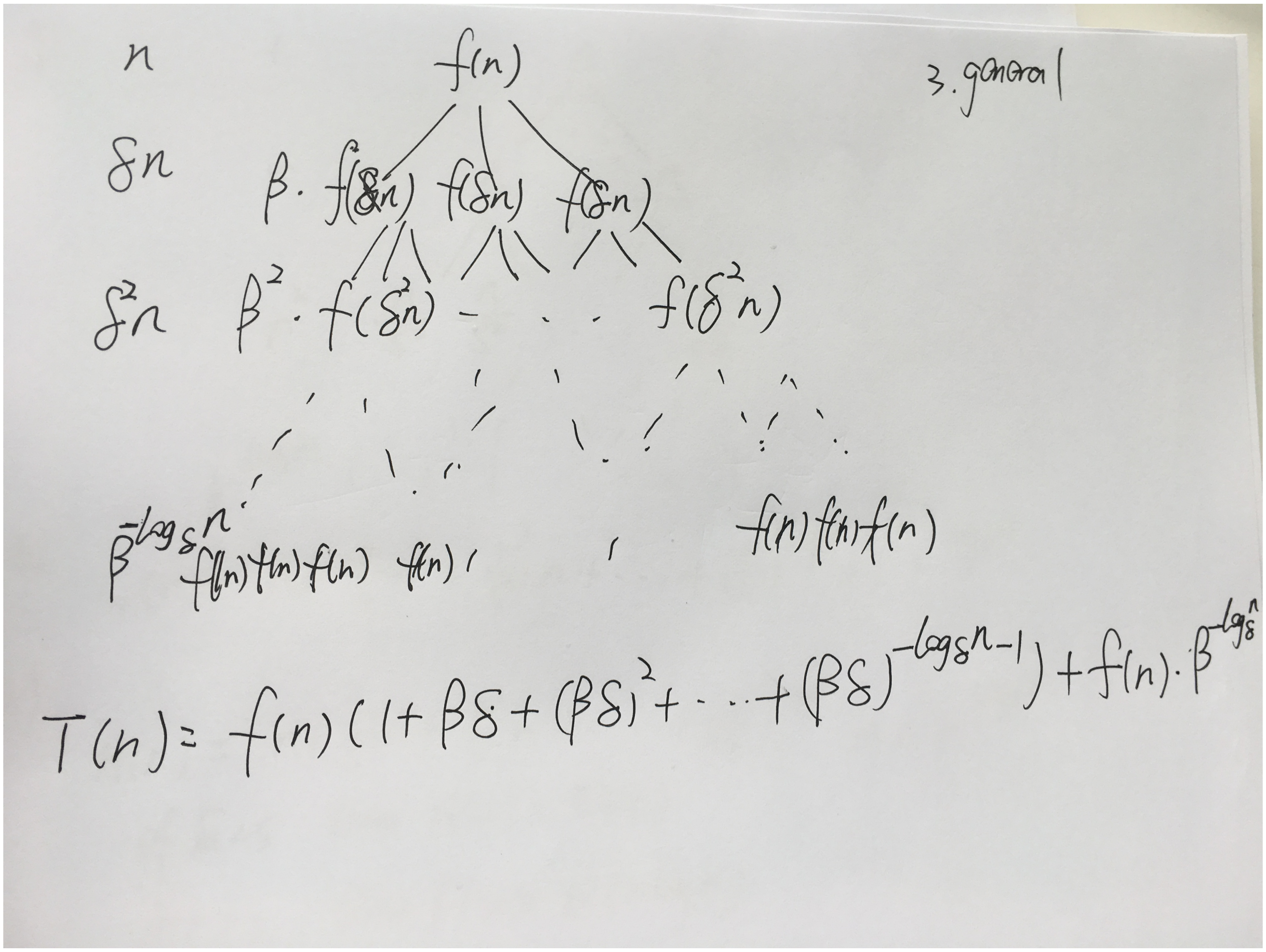


Figure 7: *Exe 3.general Solution.*