

Competitive Algorithm Design and Practice

Dynamic Programming

2014/03/19

Yi Long, Lu (mike199250)

mike199250@gmail.com

http://myweb.ncku.edu.tw/~f74991073/20140319_DP.zip

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan



Recall Divide & Conquer



Recall Divide & Conquer

- Do you remember.....

```

1  /* file name: fibonacci.c */
2  #include <stdio.h>
3  int fib(int n)
4  {
5      if(n<=1) return n;
6      else return fib(n-1)+fib(n-2);
7  }
8  int main()
9  {
10     int i;
11     for(i=0;i<10;i++)
12         printf("F%d is: %d\n",i,fib(i));
13
14     return 0;
15 }

```

```

1  /* file name: fibonacci_bottom_up.c */
2  #include <stdio.h>
3  int fib(int n)
4  {
5      int i,f0=0,f1=1,f2;
6      if(n<=1) return n;
7      for(i=2;i<=n;i++)
8      {
9          f2=f0+f1;
10
11          f0=f1;
12          f1=f2;
13      }
14      return f2;
15  }
16  int main()
17  {
18     int i;
19     for(i=0;i<10;i++)
20         printf("F%d is: %d\n",i,fib(i));
21
22     return 0;
23 }

```



Recall Divide & Conquer

- Have you tried.....

Learn more!

- 1: For **Top-down** and **Bottom-up** Fibonacci numbers, try to call `fib(45)`, what happen?
- 2: **How to use** pair, set or other STL? Maybe you can find it out **on the internet?**

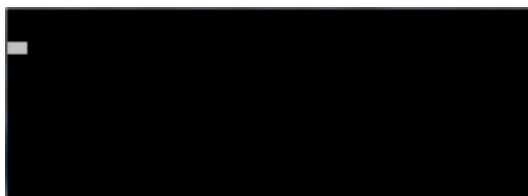
Made By mike199250



Recall Divide & Conquer

- If you tried.....

- Top-down:



```
F(45) is: 1134903170  
Process returned 0 (0x0)   execution time : 30.115 s  
Press any key to continue.
```

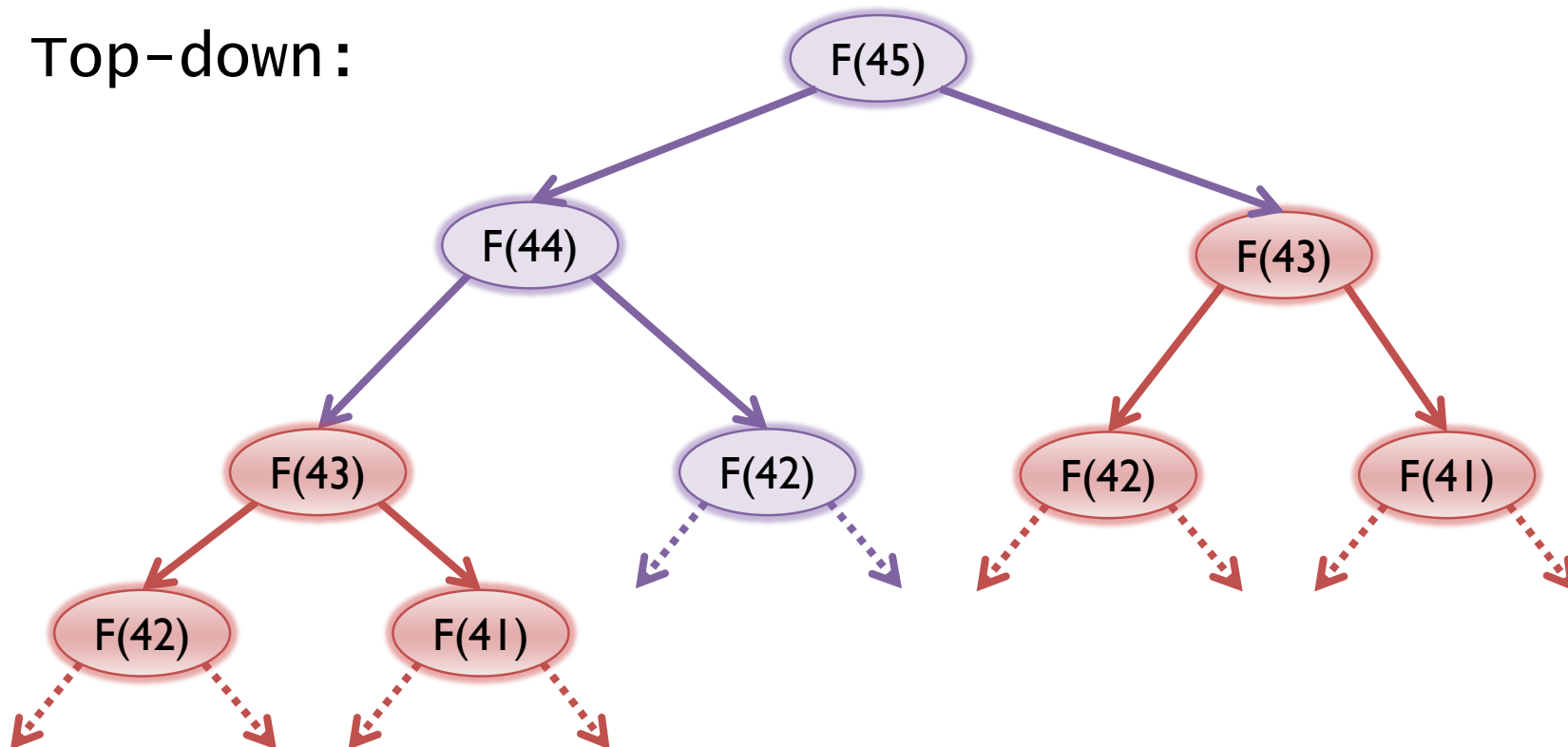
- Bottom-up:

```
F(45) is: 1134903170  
Process returned 0 (0x0)   execution time : 0.074 s  
Press any key to continue.
```



Recall Divide & Conquer

- Top-down:

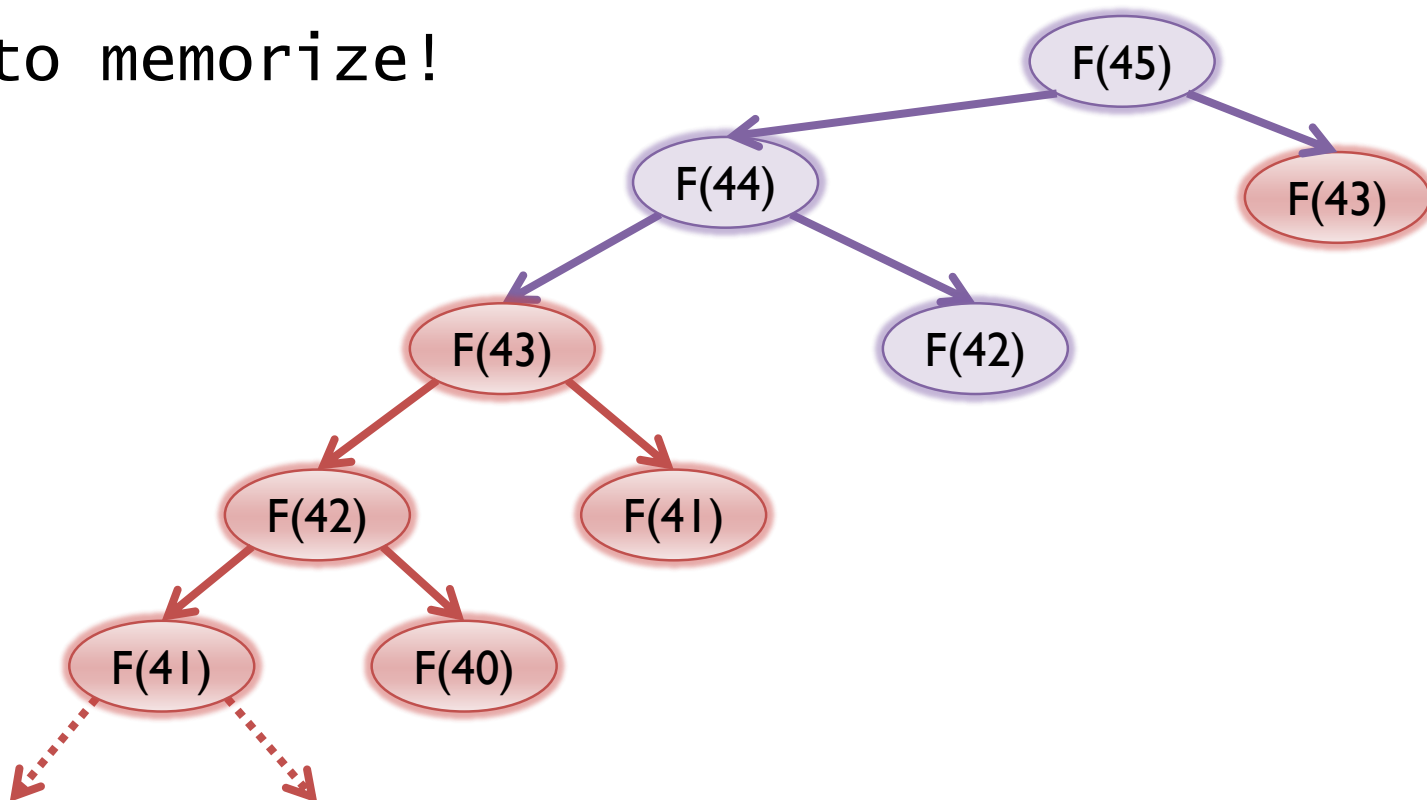


- It calculate $F(43), F(42), \dots$ many times!



Recall Divide & Conquer

- Try to memorize!



Recall Divide & Conquer

```
1  /* file name: fibonacci_dp_top_down.c */
2  #include <stdio.h>
3  #include <string.h>
4  int num[46];
5  int fib(int n)
6  {
7      if(n<=1) return n;
8      else if(num[n]!=-1) return num[n];
9      else return num[n]=fib(n-1)+fib(n-2);
10 }
11 int main()
12 {
13     memset(num,-1,sizeof(num));
14     printf("F(45) is: %d\n",fib(45));
15     return 0;
16 }
```

F(45) is: 1134903170

Process returned 0 (0x0) execution time : 0.056 s
Press any key to continue.



Dynamic Programming



Dynamic Programming

- What is DP?
- Optimal substructure.
- Overlapping sub-problems.
- bla.....bla.....bla.....
- Just **focus on** two questions:
 - **what** do we want to know?
 - **How** can we get that?
- And remember **not to recalculate**.



Dynamic Programming

- what do we want to know?
 - The Fibonacci number F_n
- How can we get that?
 - $F_n = F_{n-1} + F_{n-2}$
- And remember not to recalculate.

