Dynamic Programming

Truong Ngoc Tuan

Nội dung

- Bài toán mở đầu
- Quy hoạch động và Chia để trị
- Lược đồ chung
- Bài toán tính số Fibonaci
- Bài toán cái túi
- Bài toán dãy con có tổng lớn nhất
- Bài toán tìm xâu con chung dài nhất

Nội dung

- Bài toán mở đầu
- Quy hoạch động và Chia để trị
- Lược đồ chung
- Bài toán tính số Fibonaci
- Bài toán cái túi
- Bài toán dãy con có tổng lớn nhất
- Bài toán tìm xâu con chung dài nhất

Bài toán mở đầu

❑ Đặt vấn đề

- Chia bài toán thành các bài toán con, trong nhiều trường hợp các bài toán con khác nhau lại chứa các bài toán con hoàn toàn giống nhau.
- VD: tính số Fibonaci thứ n, F(n)

```
F(0) = 0, F(1) = 1
F(n) = F(n-2) + F(n-1) với mọi n > 1
```

Bài toán mở đầu

❑ Đặt vấn đề

- Tiếp cận theo hướng: chia để trị

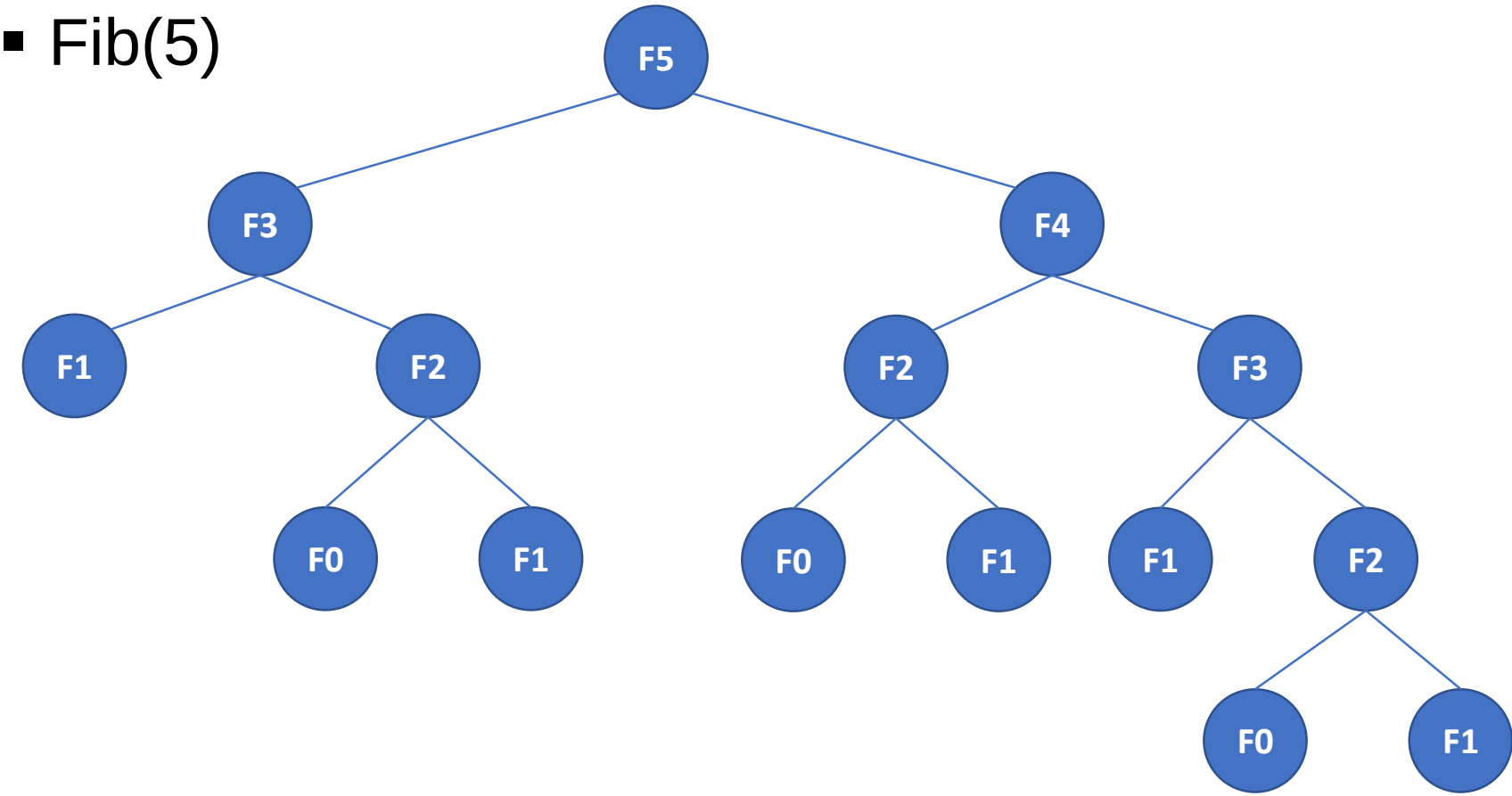
$$F(n) = F(n-1) + F(n-2)$$

```
int Fib (n) {
  if (n < 2)
      return n;
  else
      return Fib(n-2) + Fib(n-1);
}
```

Bài toán mở đầu

❑ Đặt vấn đề

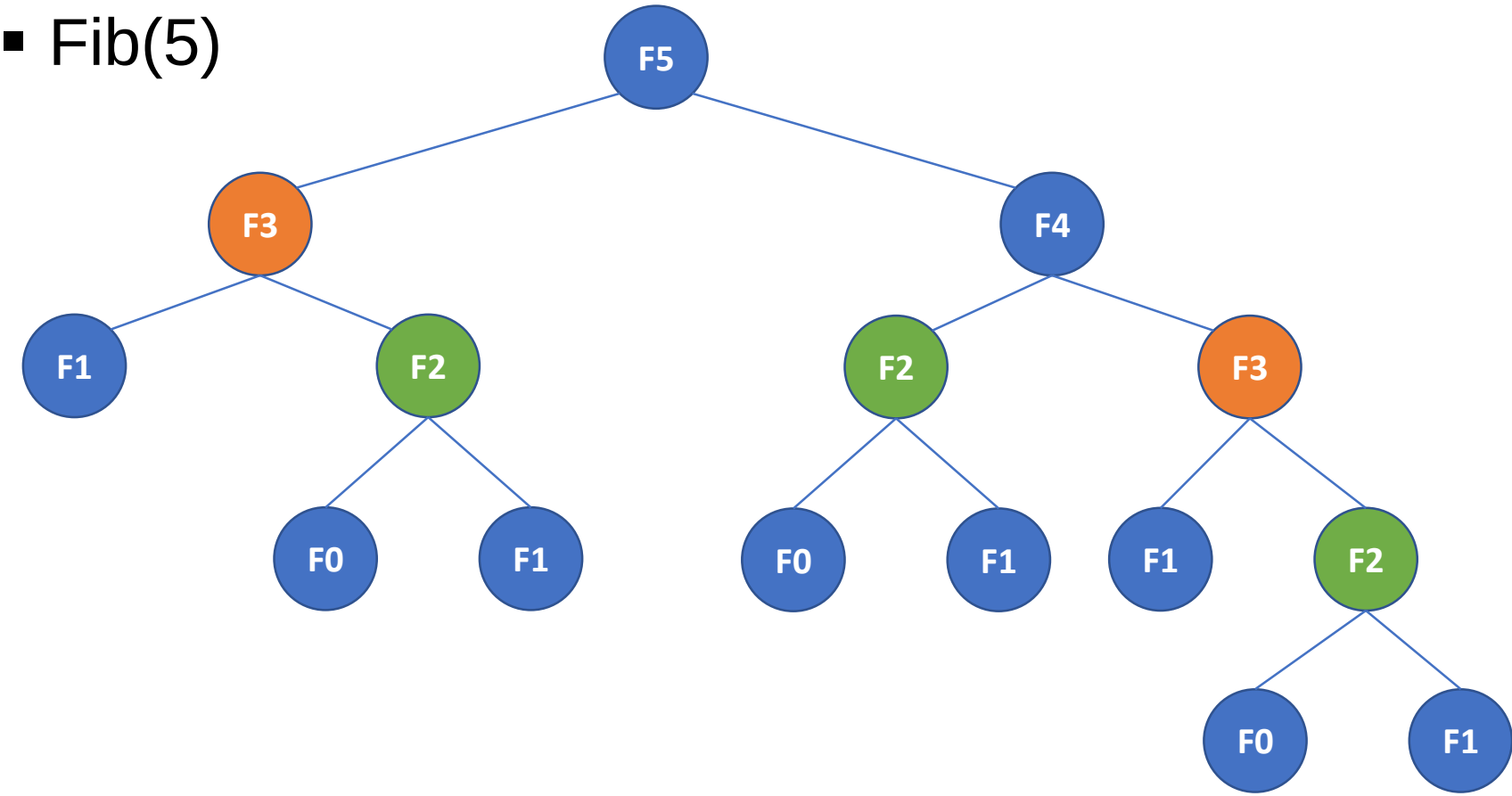
- Fib(5)



Bài toán mở đầu

❑ Đặt vấn đề

- Fib(5)



Bài toán mở đầu

❑ Nhận xét

- Tính lại các bài toán con nhiều lần
- Ảnh hưởng đến bộ nhớ và thời gian thực thi
- Cần tìm 1 giải pháp khắc phục vấn đề trên

➡ **Quy hoạch động**

Bài toán mở đầu

❑ Quy hoạch động

- Thiết kế thuật toán theo kiểu chia bài toán lớn thành các bài toán con
- Sử dụng lời giải của các bài toán con để tìm lời giải cho bài toán ban đầu
- Cần tìm 1 giải pháp khắc phục vấn đề trên

Bài toán mở đầu

❑ Quy hoạch động

- Thay vì gọi đệ quy như “chia để trị”, quy hoạch động sẽ **tính trước** lời giải của các bài toán con và lưu vào bộ nhớ (mảng)
- Sau đó lấy lời giải của bài toán con ở trong mảng đã tính trước để giải bài toán lớn

Bài toán mở đầu

❑ Quy hoạch động

- Thay vì gọi đệ quy như “chia để trị”, quy hoạch động sẽ **tính trước** lời giải của các bài toán con và lưu vào bộ nhớ (mảng)
- Sau đó lấy lời giải của bài toán con ở trong mảng đã tính trước để giải bài toán lớn

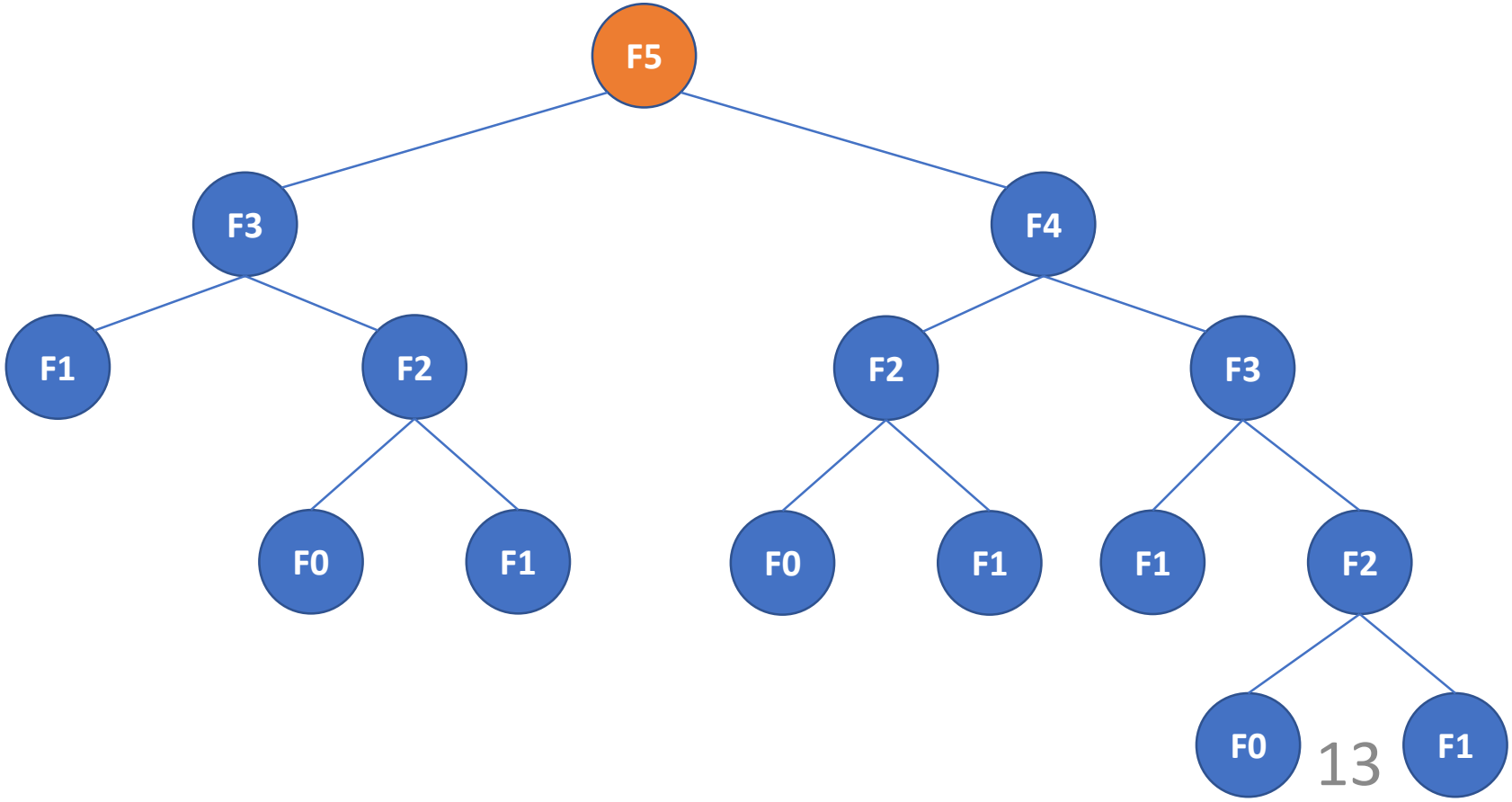
Nội dung

- Bài toán mở đầu
- Quy hoạch động và Chia để trị
- Lược đồ chung
- Bài toán tính số Fibonaci
- Bài toán cái túi
- Bài toán dãy con có tổng lớn nhất
- Bài toán tìm xâu con chung dài nhất

Quy hoạch động và Chia để trị

❑ Chia để trị

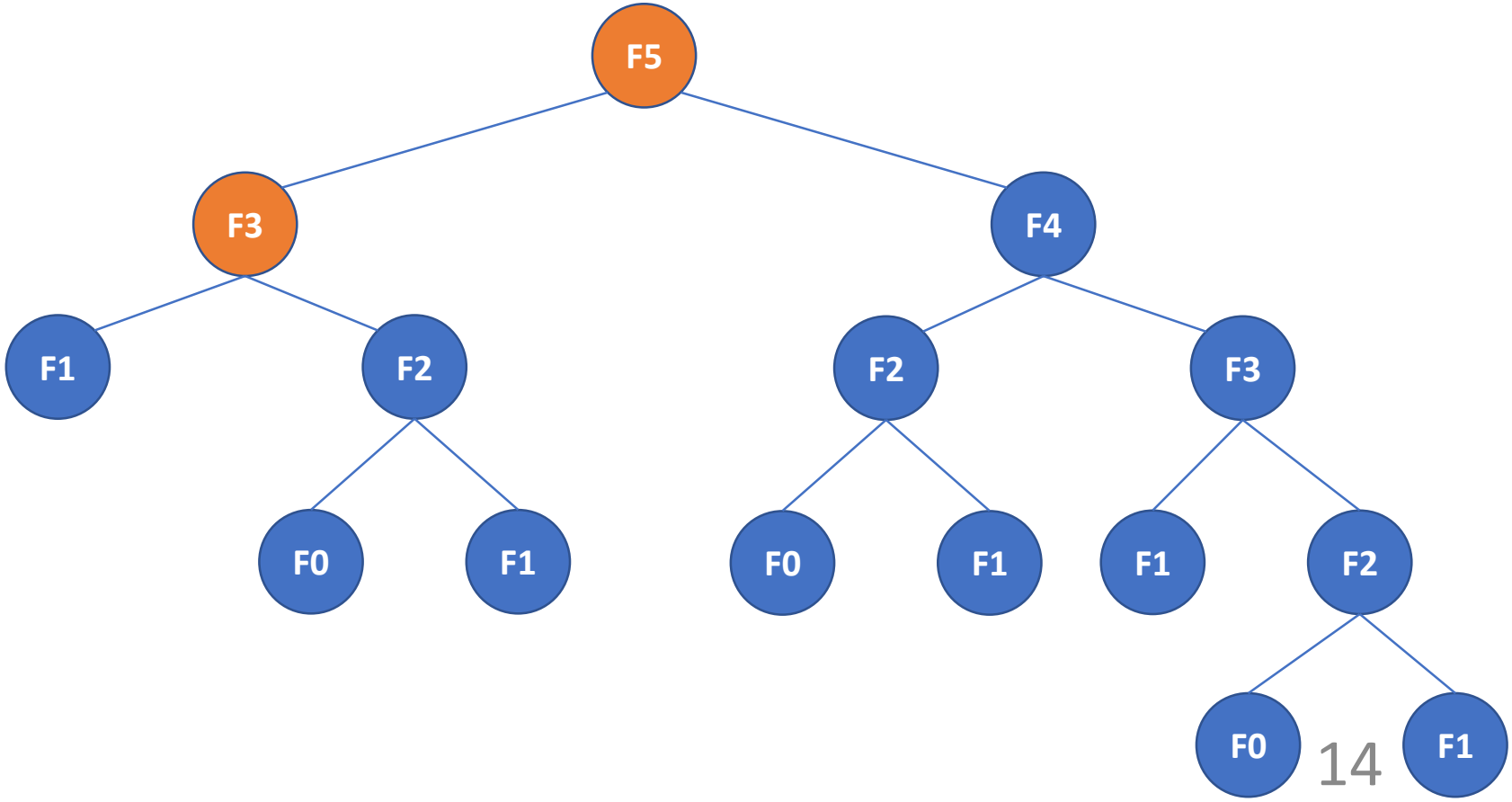
Tiếp cận từ trên xuống (Top - Down)



Quy hoạch động và Chia để trị

❑ Chia để trị

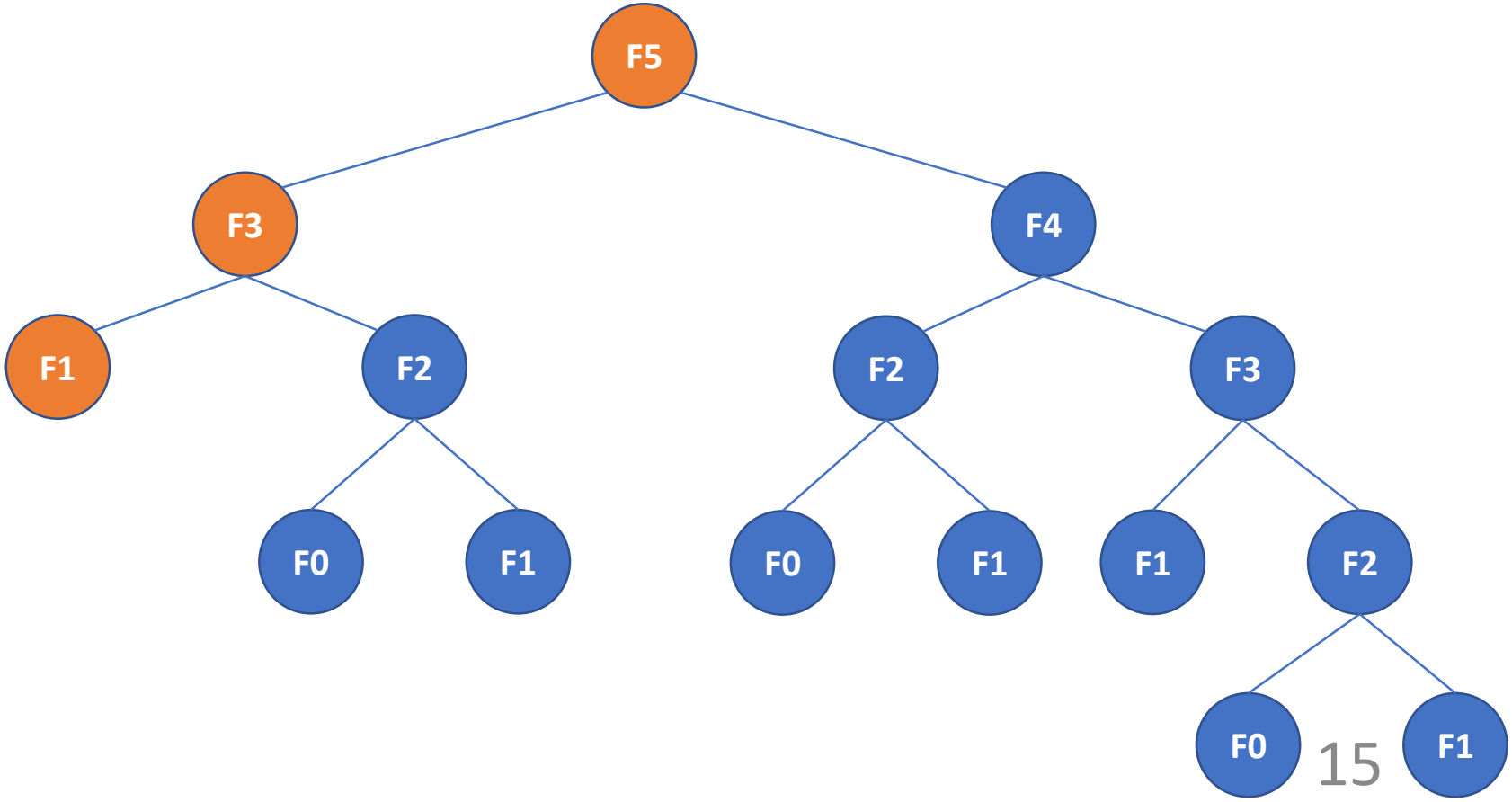
Tiếp cận từ trên xuống (Top - Down)



Quy hoạch động và Chia để trị

❑ Chia để trị

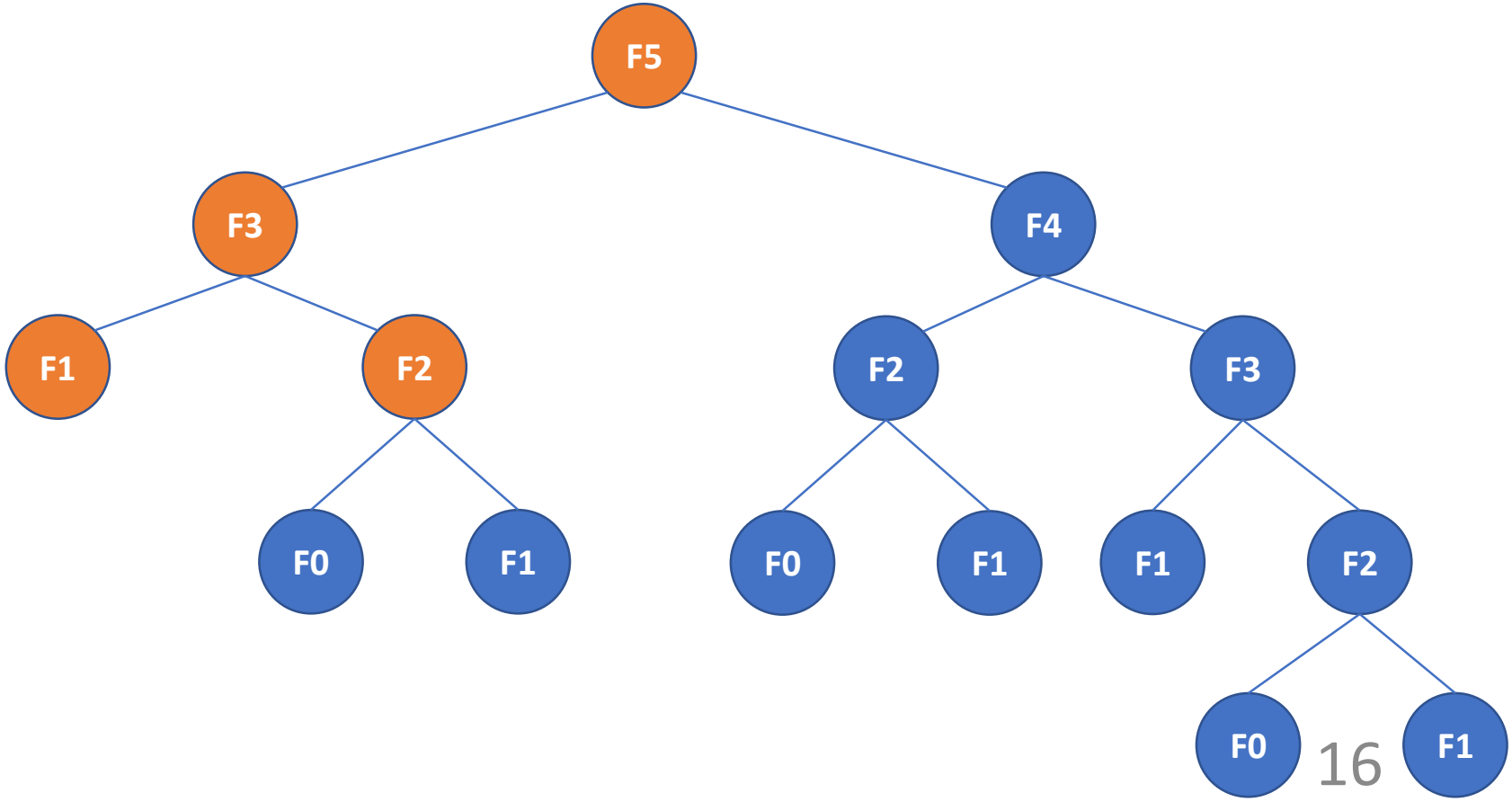
Tiếp cận từ trên xuống (Top - Down)



Quy hoạch động và Chia để trị

❑ Chia để trị

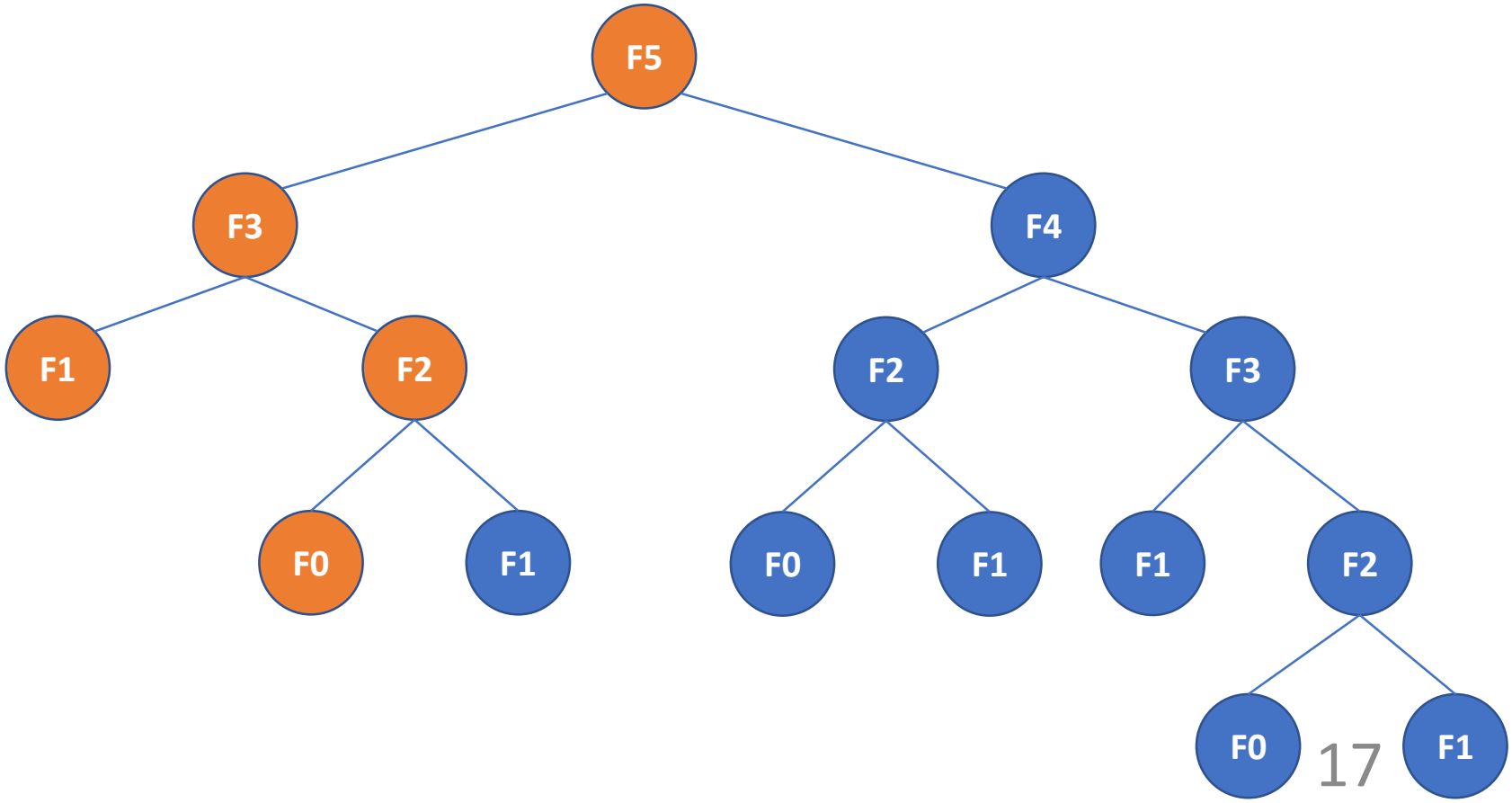
Tiếp cận từ trên xuống (Top - Down)



Quy hoạch động và Chia để trị

❑ Chia để trị

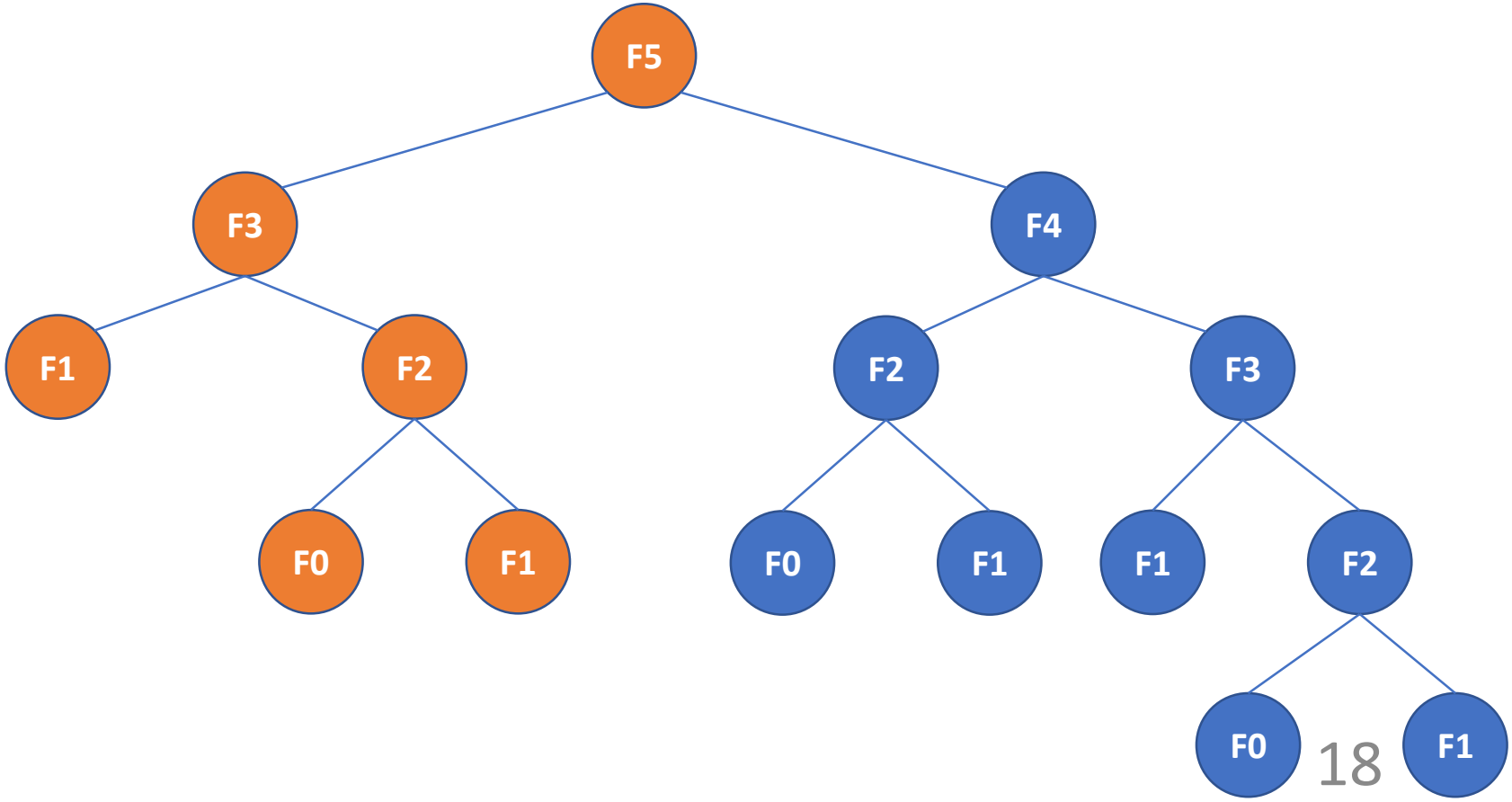
Tiếp cận từ trên xuống (Top - Down)



Quy hoạch động và Chia để trị

❑ Chia để trị

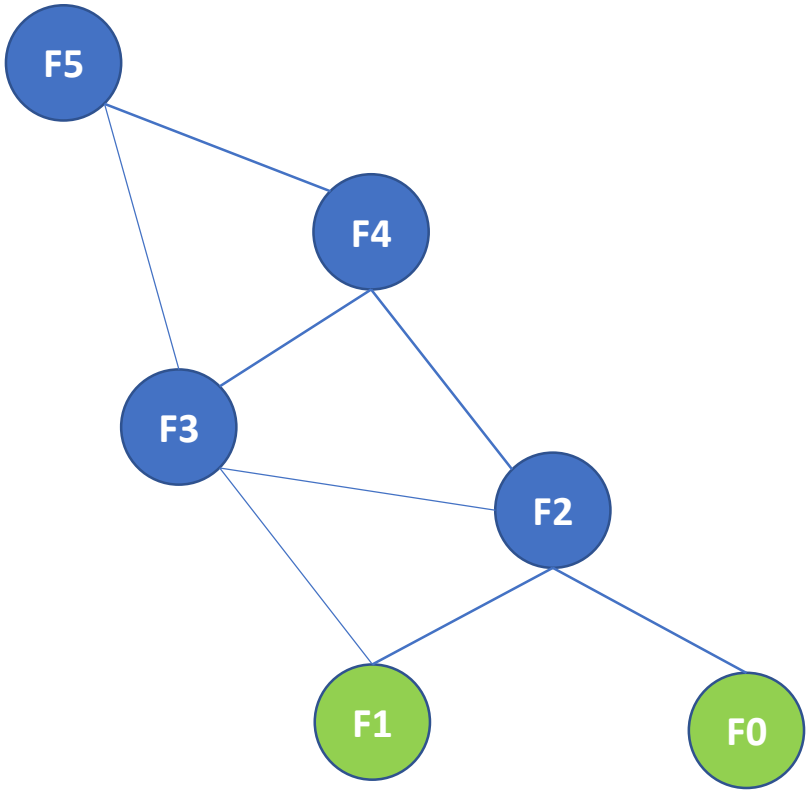
Tiếp cận từ trên xuống (Top - Down)



Quy hoạch động và Chia để trị

❑ Quy hoạch động

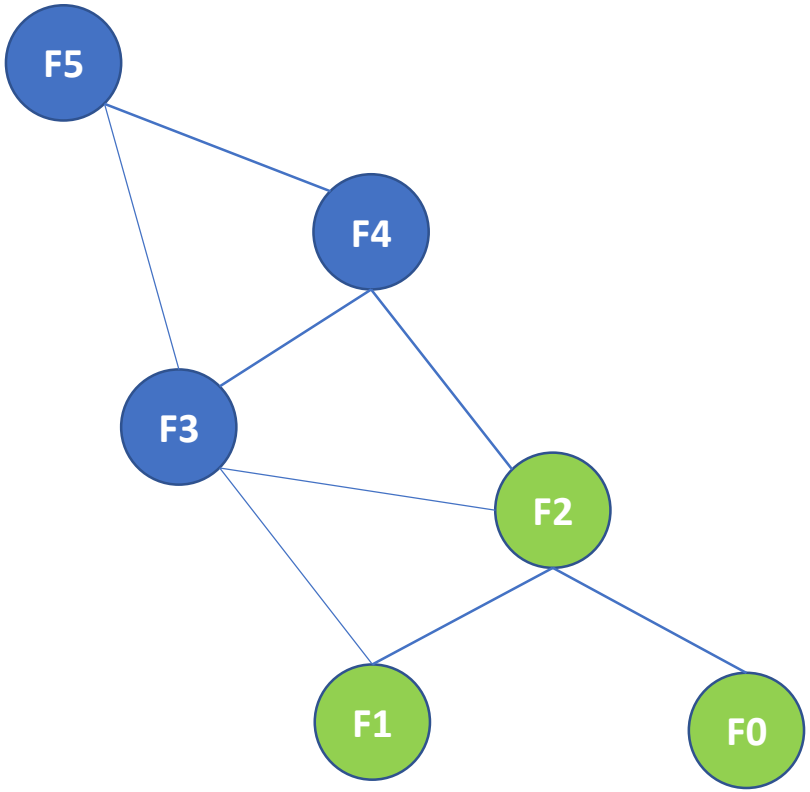
Tiếp cận từ dưới lên (Bottom - Up)



Quy hoạch động và Chia để trị

❑ Quy hoạch động

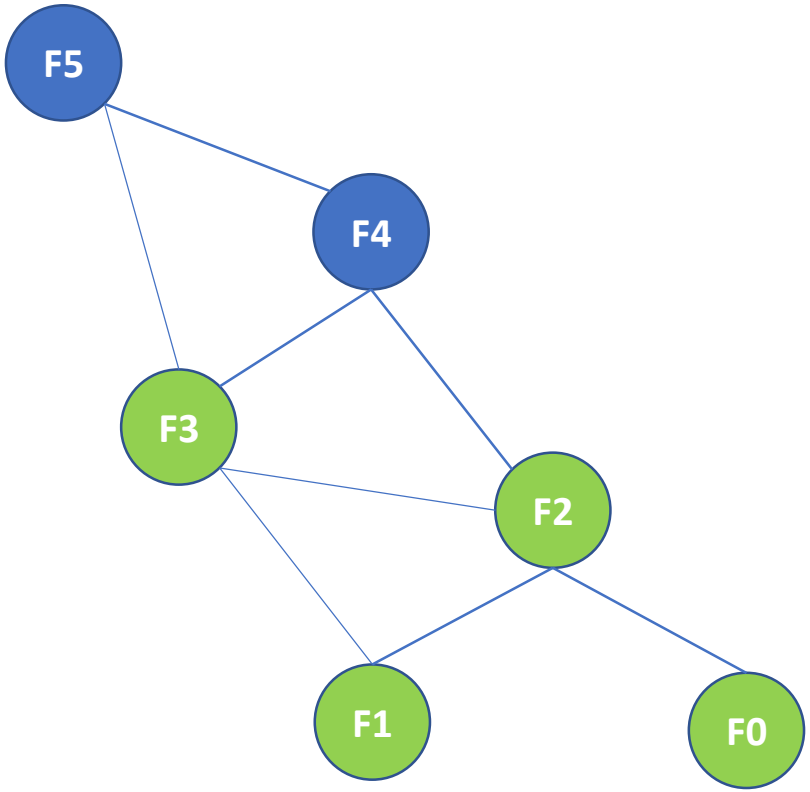
Tiếp cận từ dưới lên (Bottom - Up)



Quy hoạch động và Chia để trị

❑ Quy hoạch động

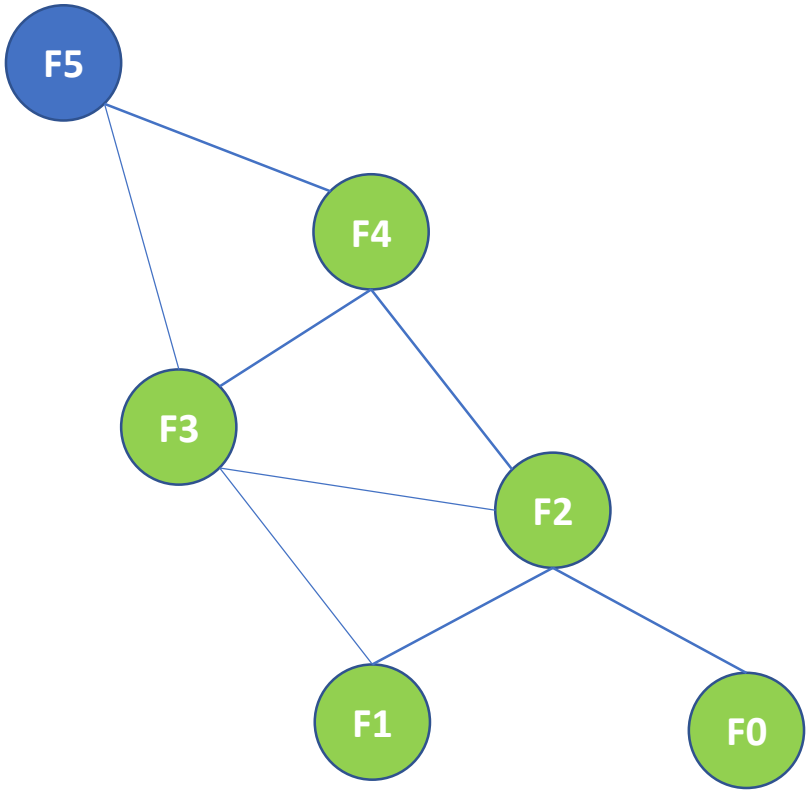
Tiếp cận từ dưới lên (Bottom - Up)



Quy hoạch động và Chia để trị

❑ Quy hoạch động

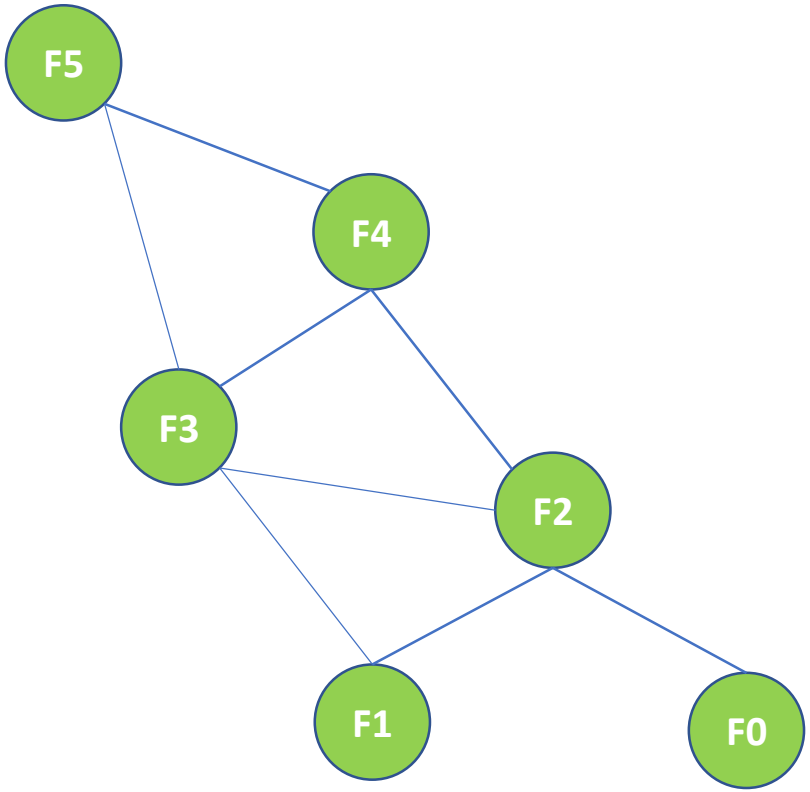
Tiếp cận từ dưới lên (Bottom - Up)



Quy hoạch động và Chia để trị

❑ Quy hoạch động

Tiếp cận từ dưới lên (Bottom - Up)



Nội dung

- Lược đồ chung
- Quy hoạch động và Chia để trị
- Lược đồ chung
- Bài toán tính số Fibonaci
- Bài toán cái túi
- Bài toán dãy con có tổng lớn nhất
- Bài toán tìm xâu con chung dài nhất

Lược đồ chung

❑ Các bước xây dựng

- Phân rã bài toán
- Giải bài toán con và ghi nhận lời giải
- Tổng hợp lời giải

Lược đồ chung

- **Phân rã bài toán**
  - Chia bài toán cần giải thành những bài toán con nhỏ hơn
  - Chia đến mức có thể giải trực tiếp được

Lược đồ chung

- Giải bài toán con và ghi nhận lời giải
  - Lưu trữ lời giải của bài toán con để sử dụng về sau
  - Thường sử dụng mảng để lưu trữ

Lược đồ chung

❑ Tổng hợp lời giải

- Tổng hợp lời giải các bài toán con kích thước nhỏ hơn thành lời giải bài toán lớn hơn
- Tiếp tục cho đến khi thu được lời giải bài toán ban đầu (bài toán có kích thước lớn nhất)

Nội dung

- Lược đồ chung
- Quy hoạch động và Chia để trị
- Lược đồ chung
- Bài toán tính số Fibonaci
- Bài toán cái túi
- Bài toán dãy con có tổng lớn nhất
- Bài toán tìm xâu con chung dài nhất

Bài toán Fibonaci

❑ Tính phần tử thứ n của dãy Fibonaci

- Phân rã

$$F(n) = F(n-1) + F(n-2)$$

- Giải bài toán con

$$F(0) = 0$$

$$F(1) = 1$$

- Tổng hợp

$$F(n) = F(n-1) + F(n-2)$$

Bài toán Fibonaci

- Cài đặt

```
int Fibonaci_DP(int n) {
  F[0] = 0; F[1] = 1;
  if (n > 1) {
      for (i: 2 → n) {
          F[i] = F[i-1] + F[i-2];
      }
  }
  return F[n];
}
```

Bài toán Fibonaci

- Cài đặt

```
int Fibonaci_DP2(int n) {
  Fi2 = 0; Fi1 = 1; i = 2;
  while (i ≤  n) {
      temp = Fi1;
      Fi1 = Fi1 + Fi2;
      Fi2 = temp;
      i++
  }
  return F[n];
}
```

Nội dung

- Lược đồ chung
- Quy hoạch động và Chia để trị
- Lược đồ chung
- Bài toán tính số Fibonaci
- Bài toán cái túi
- Bài toán dãy con có tổng lớn nhất
- Bài toán tìm xâu con chung dài nhất

Bài toán cái túi

❑ Knapsack Problem

- Có n đồ vật, đồ vật thứ i có trọng lượng w_i và giá trị v_i (i: 1 → n)
- Tìm cách lấy các đồ vật này cho vào túi có dung lượng m sao cho:
  - Tổng trọng lượng các đồ vật cho vào túi không vượt quá m
  - ***Tổng giá trị của các đồ vật là lớn nhất***

Bài toán cái túi

❑ Knapsack Problem

- Có n đồ vật, đồ vật thứ i có trọng lượng w_i và giá trị v_i (i: 1 → n)
- Tìm cách lấy các đồ vật này cho vào túi có dung lượng m

Phương pháp Tham lam

Kết quả nhận được thường là không tối ưu

Bài toán cái túi

❑ Knapsack Problem

- Có n - đồ vật, m – trọng lượng túi
- ***Phân rã:***
  - Gọi tập chỉ số các đồ vật là i (1..n)
  - Gọi khối lượng túi là L (0..m)
  - Gọi MaxV(i, L) là tổng giá trị lớn nhất có thể chọn trọng i đồ vật (1..i) với trọng lượng túi tối đa là L
  - Khi đó MaxV(n, m) là giá trị lớn nhất có thể mang đi được

Bài toán cái túi

❑ Knapsack Problem

- Có n - đồ vật, m – trọng lượng túi
- ***Giải bài toán con:***
  - MaxV(0, L) = 0 với mọi L
  - MaxV(i, 0) = 0 với mọi i

Bài toán cái túi

❑ Knapsack Problem

- Có n - đồ vật, m – trọng lượng túi
- ***Tổng hợp:***
  - Đã có MaxV(i-1, L): giá trị lớn nhất mang đi được với i-1 đồ vật khi trọng lượng túi là L

Bài toán cái túi

❑ Knapsack Problem

- Có n - đồ vật, m – trọng lượng túi
- ***Tổng hợp:***
  - Xét đồ vật thứ i khi trọng lượng túi vẫn là L
    - Chỉ mang them đồ vật i khi giá trị của túi lúc mang đồ vật i-1 là: L – w[i]
    - Khi mang đồ vật i, giá trị túi lớn hơn không mang đồ vật i: MaxV(i-1, L)

Bài toán cái túi

❑ Knapsack Problem

- Có n - đồ vật, m – trọng lượng túi
- ***Tổng hợp:***
  - Công thức tổng quát

MaxV(i, L) = MAX {

MaxV(i-1, L- w[i]) + v[i],

MaxV(i-1, L)

}

Bài toán cái túi

❑ Cài đặt

```
int knapsack_DP() {
   for L: 0 → m: MaxV[0, L]     = 0;
   for i: 0 → n: MaxV[i, 0]     = 0;
   for i: 1 → n:
        for L: 1 → m:
             MaxV[i, L] = MaxV[i-1, L];
             if (L ≥ w[i] &&
               (MaxV[i-1, L-w[i]] + v[i] > MaxV[i-1, L])
                  MaxV[i, L] = MaxV[i-1, L-w[i]] + v[i];
   return MaxV(n, m);
}
```

Bài toán cái túi

❑ Minh hoạ

- Cho 6 đồ vật (n=6) và trọng lượng túi m=19.

Các đồ vật có trọng lượng và giá trị như sau

i	W	V
1	3	7
2	4	10
3	5	20
4	7	19
5	6	13
6	9	40

Nội dung

- Lược đồ chung
- Quy hoạch động và Chia để trị
- Lược đồ chung
- Bài toán tính số Fibonaci
- Bài toán cái túi
- Bài toán dãy con có tổng lớn nhất
- Bài toán tìm xâu con chung dài nhất

Bài toán dãy con có tổng lớn nhất

❑ Bài toán

- Cho mảng gồm N phần tử: A[1..N]
- Hãy tìm dãy con các phần tử liên tiếp của A có tổng lớn nhất
- ***VD:*** ***13, -15, 2, 18, 4, 8, 0, -5, -8***
- Dãy con cần tìm là:

13, -15, 2, 18, 4, 8, 0, -5, -8

Bài toán dãy con có tổng lớn nhất

❑ Ý tưởng – Quy hoạch động

- Phân rã
  - Gọi ***MaxS[i]*** là tổng lướn nhất của dãy con liên tiếp có i phần tử a[1] → a[i]
  - Khi đó ***MaxS[N]*** là giá trị lớn nhất của dãy con liên tiếp cần tìm

Bài toán dãy con có tổng lớn nhất

❑ Ý tưởng – Quy hoạch động

- Bài toán cơ sở
  - Với i = 1 ta có MaxS[i] = a[i]

Bài toán dãy con có tổng lớn nhất

❑ Ý tưởng – Quy hoạch động

- Tổng hợp
  - Giả sử i > 1 và MaxS[k] là giá trị lớn nhất đã biết với k: 1 → i-1
  - Ta cần tính MaxS[i] là tổng dãy con liên tiếp lớn nhất của dãy a[1],…, a[i-1], a[i]
  - Các dãy con liên tiếp của dãy này có thể là:
    - Các dãy con liên tiếp có chứa a[i]
    - Các dãy con liên tiếp không chứa a[i]

Bài toán dãy con có tổng lớn nhất

❑ Ý tưởng – Quy hoạch động

- Tổng hợp
  - Gọi ***MaxE[i]*** là tổng lớn nhất của các dãy con liên tiếp của dãy a[1],…, a[i] có chứa a[i]
  - Tổng lớn nhất của các dãy con liên tiếp của dãy a[1],…, a[i] nếu không chứa a[i] là tổng của các dãy con của dãy a[1],…, a[i-1] hay ***MaxS[i-1]***

➡ ***MaxS[i] = MAX(MaxS[i-1], MaxE[i])***

Bài toán dãy con có tổng lớn nhất

❑ Ý tưởng – Quy hoạch động

- Tính ***MaxE[i]***
  - Với i = 1 thì MaxE[i] = a[1]
  - Với i > 1:

$$MaxE[i] = \begin{cases} MaxE[i-1] + a[i] \\ a[i] \end{cases}$$

➡ ***MaxE[i] = MAX(a[i], MaxE[i-1] + a[i]), i>1***

Bài toán dãy con có tổng lớn nhất

- Cài đặt

```
void sub_max(int a[]) {
    MaxS = a[1];        MaxE = a[1];
    start = 1;          end = 1;            temp_start = 1;
    for (int i=2; i <= n; i++){
          if (MaxE > 0 )
                MaxE = MaxE + a[i];
          else {
                MaxE = a[i];
                temp_start = i;
          }
          if (MaxE > MaxS) {
                MaxS = MaxE;
                end = i; start = temp_start;
          }
    }
}
```

Bài toán dãy con có tổng lớn nhất

- Minh hoạ
  - Dãy a gồm 9 phần tử

    a = {13, -15, 2, 18, 4, 8, 0, -5, -8}

Nội dung

- Lược đồ chung
- Quy hoạch động và Chia để trị
- Lược đồ chung
- Bài toán tính số Fibonaci
- Bài toán cái túi
- Bài toán dãy con có tổng lớn nhất
- Bài toán tìm xâu con chung dài nhất

Bài toán tìm xâu con chung dài nhất

❑ Bài toán

- Cho 2 xâu

$$X = (x_1, x_2, \ldots, x_m)$$

$$Y = (y_1, y_2, \ldots, y_n)$$

- Hãy tìm xâu con chung dài nhất của 2 xâu X, Y
- Ví dụ

X = “KHOA HOC”

Y = “HOA HONG” → **HOA HO**

Bài toán tìm xâu con chung dài nhất

❑ Ý tưởng – Quy hoạch động

- Phân rã
  - m – chiều dài xâu X, n – chiều dài xâu Y
  - Gọi L[i, j] là độ dài dãy con chung dài nhất của 2 dãy

$$\mathbf{X_i} = x_1\, x_2 \ldots x_i \qquad \text{và} \qquad \mathbf{Y_j} = y_1\, y_2 \ldots y_j$$

  - Khi đó L[m, n] là độ dài xâu con chung dài nhất của X và Y

Bài toán tìm xâu con chung dài nhất

- Ý tưởng – Quy hoạch động
  - Bài toán con
    - L[0, j] = 0 với $\forall j: 1 \rightarrow n$
    - L[i, 0] = 0 với $\forall i: 1 \rightarrow m$

Bài toán tìm xâu con chung dài nhất

❑ Ý tưởng – Quy hoạch động

- Tổng hợp
  - L[i, j] = L[i-1, j-1] + 1 với $x_i = y_i$
  - L[i, j] = MAX(L[i-1, j], L[i, j-1]) với $x_i \neq y_i$

Bài toán tìm xâu con chung dài nhất

- Cài đặt

```
void LCS (X, Y) {
    for (i: 1→m)      L[i, 0] = 0;
    for (j: 1→n) L[0, j] = 0;

    for (i: 1→m)
        for (i: 1→n)
            if (x_i = y_i)
                L[i, j] = L[i-1, j-1] + 1;
            else
                L[i, j] = MAX (L[i-1, j], L[i, j-1] );
}
```


https://vnoi.info/wiki/algo/dp/basic-problems#x%C3%A2u-con-chung-d%C3%A0i-nh%E1%BA%A5t


Q
A


THANK
YOU