

Dynamické programování

- Využití zejména pro zefektivnění výpočtů, kde se něco náročného vyhodnocuje opakovaně.
- Náročnost výpočtu snížíme využitím pomocné paměti.

Dynamické programování (2)

- Začínáme u zápisu algoritmu tak, jak je věc definována – s rekurzí a mnohonásobným vyhodnocováním téhož.
- Pak si zkusíme pamatovat výsledky výpočtů, které jsme již provedli
- Nakonec se pokusíme najít takový výpočet, který eliminuje alespoň část zbytečných volání – nejlépe takový, který to bude počítat přímo.

Příklad: prvky Fibonacciho posloupnosti

- Nejprve si připomeneme definici,
- pak zakódujeme výpočet dle definice,
- potom zrealizujeme meziukládání,
- nakonec naplníme vyrovnávací paměť co nejefektivněji.

Příklad: prvky Fibonacciho posloupnosti

- $F_0 = F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$ pro $n \geq 2$

Příklad: prvky Fibonacciho posloupnosti (2)

```
FUNCTION Fib(n:Integer):Longint;  
  BEGIN  
    IF n<0 THEN  
      BEGIN  
        WriteLn('Fib: negative numbers not accepted.');        Halt  
      END;  
    IF n<2 THEN Fib:=1  
      ELSE Fib:=Fib(n-1)+Fib(n-2)  
    END;  
  END;
```

Příklad: prvky Fibonacciho posloupnosti (3)

```
CONST
```

```
    Max = 45;
```

```
VAR
```

```
    buffer : ARRAY[0..Max] OF Longint;
```

```
PROCEDURE BufInit;
```

```
    VAR
```

```
        i : Integer;
```

```
    BEGIN
```

```
        FOR i:=0 TO Max DO buffer[i]:=0;
```

```
    END;
```

Příklad: prvky Fibonacciho posloupnosti (4)

```
FUNCTION Fib2(n:integer):Longint;  
  BEGIN  
    IF (n<0)OR(n>Max) THEN  
      BEGIN  
        WriteLn('Fib2: Out of range.');
```

Halt

```
      END;  
    IF buffer[n]=0 THEN  
      IF n<2 THEN buffer[n]:=1  
        ELSE buffer[n]:=Fib2(n-1)+Fib2(n-2);  
    Fib2:=buffer[n]  
  END;
```

Příklad: prvky Fibonacciho posloupnosti (5)

```
VAR
```

```
  Fib3buf : ARRAY[0..Max] OF Longint;
```

```
PROCEDURE Fib3init;
```

```
  VAR
```

```
    i : Integer;
```

```
  BEGIN
```

```
    Fib3buf[0]:=1; Fib3buf[1]:=1;
```

```
    FOR i:=2 TO Max DO
```

```
      Fib3buf[i]:=Fib3buf[i-1]+Fib3buf[i-2];
```

```
  END;
```


Příklad: prvky Fibonacciho posloupnosti (6)

```
FUNCTION Fib3(n:integer):Longint;  
  BEGIN  
    IF (n<0)OR(n>Max) THEN  
      BEGIN  
        WriteLn('Fib3: Out of range.');
```

Halt

```
      END;  
      Fib3:=buffer[n]  
    END;
```