

Another variant: Bottom-up

Bottom-up

- Fibonacci: More «natural» way:
- To get the n^{th} number: Calc 1^{st} , 2^{nd} , ... $\rightarrow n^{\text{th}}$
- Linear too!

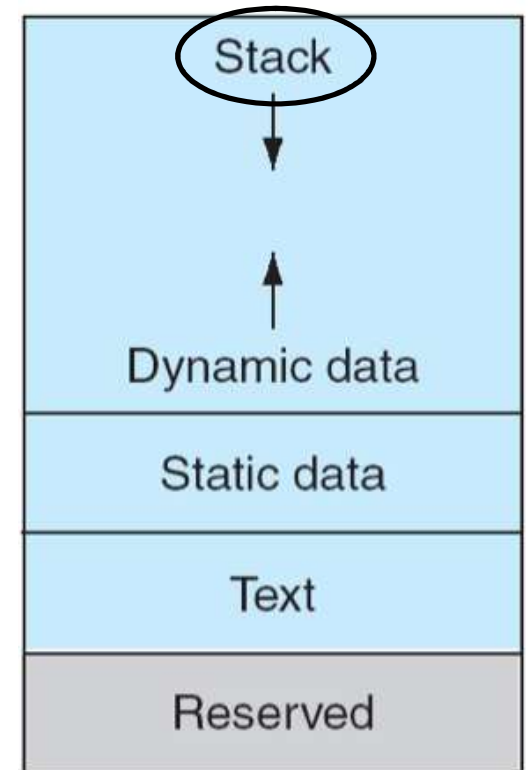
Bottom-up: Fibonacci implementation

Bottom-up

- Similarities to Memoization:
 - Array of length n that eventually is filled with $\text{fib}(1) \dots \text{fib}(n)$
- Differences:
 - Not recursively
 - Array gets filled from the **bottom-up** (not from the top)

Bottom-up: Advantages

- Advantages:
 - Can save memory
 - Faster by a constant
 - No stack limit exceeded
- Difficulties:
 - Must know order of calculations



Bottom-up: Bottles implementation

Bytelandian coins: Addition

- Time complexity?
 - Memory complexity linear -> Time complexity linear too
- Use map and set -> Runtime better than linear
- Not achievable by bottom-up

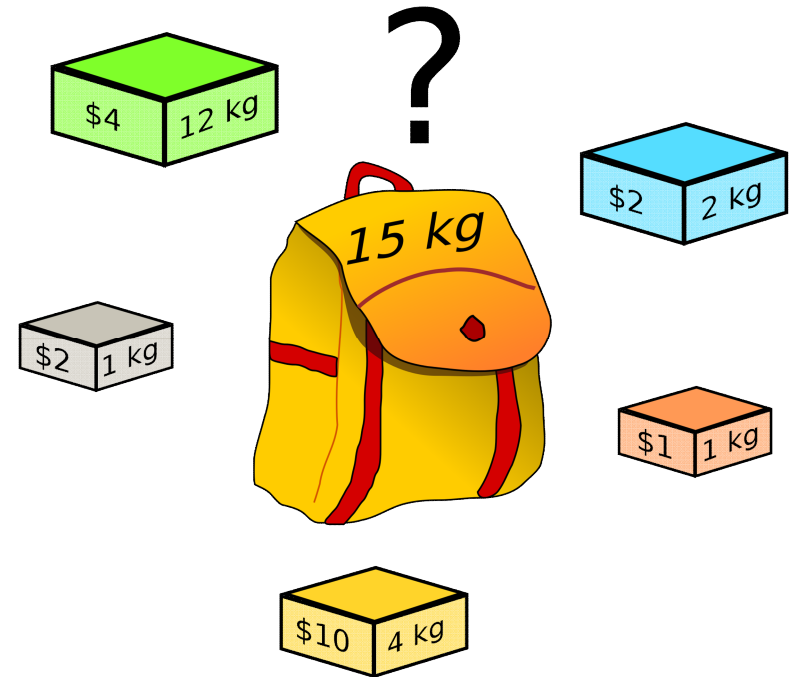
Bottom-up: Conclusion

- Useful to know
- Concentrate on learning Memoization

Multiple Dimensions

The Knapsack Problem

- n items: value and weight
- Limited capacity c in knapsack
- Which ones to take?



Knapsack Implementation