Lesson 11. Formulating Dynamic Programming Recursions

1 Formulating DP recursions

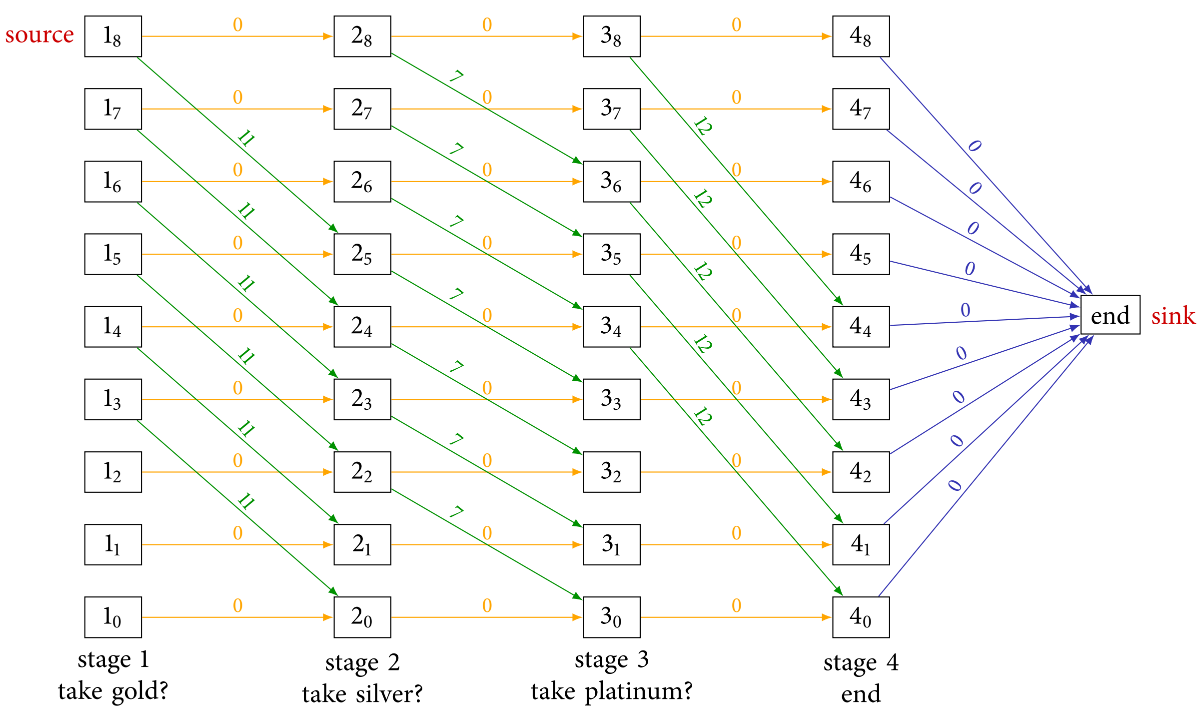
- Last lesson: recursions for shortest path problems
- Dynamic programs are not usually given as shortest/longest path problems
  - However, it is usually easier to think about DPs this way
- Instead, the standard way to describe a dynamic program is a recursion
- Let's revisit the knapsack problem that we studied back in Lesson 5 and formulate it as a DP recursion

Example 1. You are a thief deciding which precious metals to steal from a vault:

	Metal	Weight (kg)	Value
1	Gold	3	11
2	Silver	2	7
3	Platinum	4	12

You have a knapsack that can hold at most 8kg. If you decide to take a particular metal, you must take all of it. Which items should you take to maximize the value of your theft?

- We formulated the following dynamic program for this problem by giving the following longest path representation:



- Let's formulate this as a dynamic program, but now by giving its recursion representation

- Let

$$w_t = \text{weight of metal } t \qquad v_t = \text{value of metal } t \qquad \text{for } t = 1, 2, 3$$

- Stages:

$$\text{stage } t \longleftrightarrow \begin{cases} \text{consider taking metal } t & \text{if } t = 1, 2, 3 \\ \text{end of process} & \text{if } t = 4 \end{cases}$$

- States: state n $\longleftrightarrow$ n kg remaining in knapsack for $n = 0, 1, \ldots, 8$

- Allowable decisions x_t at stage t and state n:

don't take metal t — take metal t

$\underline{t = 1, 2, 3}$: x_t must satisfy $x_t \in \{0, 1\}$

$n \geq w_t x_t$ — we can take metal t only if we have enough capacity

$\underline{t = 4}$: no decisions

- Reward of decision x_t at stage t and state n:

$$v_t x_t = \begin{cases} v_t & \text{if } x_t = 1 \text{ (take metal } t\text{)} \\ 0 & \text{o/w} \end{cases} \qquad \text{for } t = 1, 2, 3, \; n = 0, 1, \ldots, 8$$

- Reward-to go function $f_t(n)$ at stage t and state n:

$f_t(n)$ = maximum value of the knapsack w/capacity n and metals $t, t+1, \ldots$ remaining — for $t = 1, 2, 3, 4$, $n = 0, 1, \ldots, 8$

- Boundary conditions: $f_4(n) = 0$ for $n = 0, 1, \ldots, 8$

- Recursion:

$$f_t(n) = \max_{\substack{x_t \in \{0,1\} \\ w_t x_t \leq n}} \left\{ v_t x_t + f_{t+1}(n - w_t x_t) \right\} \qquad \text{for } t = 1, 2, 3, \; n = 0, 1, \ldots, 8$$

- Desired reward-to-go function value: $f_1(8)$

- In general, to formulate a DP with its recursive representation:

Dynamic program – recursive representation

- **Stages** $t = 1, 2, \ldots, T$ and **states** $n = 0, 1, 2, \ldots, N$
- Allowable **decisions** x_t at stage t and state n $\quad (t = 1, \ldots, T-1;\ n = 0, 1, \ldots, N)$
- **Cost/reward** of decision x_t at stage t and state n $\quad (t = 1, \ldots, T;\ n = 0, 1, \ldots, N)$
- **Cost/reward-to-go** function $f_t(n)$ at stage t and state n $\quad (t = 1, \ldots, T;\ n = 0, 1, \ldots, N)$
- **Boundary conditions** on $f_T(n)$ at state n $\quad (n = 0, 1, \ldots, N)$
- **Recursion** on $f_t(n)$ at stage t and state n $\quad (t = 1, \ldots, T-1;\ n = 0, 1, \ldots, N)$

$$f_t(n) = \min_{x_t \text{ allowable}} \text{or} \max \left\{ \begin{pmatrix} \text{cost/reward of} \\ \text{decision } x_t \end{pmatrix} + f_{t+1} \begin{pmatrix} \text{new state} \\ \text{resulting} \\ \text{from } x_t \end{pmatrix} \right\}$$

- **Desired cost-to-go function value**

- How does the recursive representation relate to the shortest/longest path representation?

Shortest/longest path		Recursive
node t_n	$\leftrightarrow$	state n at stage t
edge $(t_n, (t+1)_m)$	$\leftrightarrow$	allowable decision x_t in state n at stage t that results in being in state m at stage $t+1$
length of edge $(t_n, (t+1)_m)$	$\leftrightarrow$	cost/reward of decision x_t in state n at stage t that results in being in state m at stage $t+1$
length of shortest/longest path from node t_n to end node	$\leftrightarrow$	cost/reward-to-go function $f_t(n)$
length of edges (T_n, end)	$\leftrightarrow$	boundary conditions $f_T(n)$
shortest or longest path	$\leftrightarrow$	recursion is min or max: $f_t(n) = \min_{x_t \text{ allowable}} \text{or} \max \left\{ \begin{pmatrix} \text{cost/reward of} \\ \text{decision } x_t \end{pmatrix} + f_{t+1} \begin{pmatrix} \text{new state} \\ \text{resulting} \\ \text{from } x_t \end{pmatrix} \right\}$
source node 1_n	$\leftrightarrow$	desired cost-to-go function value $f_1(n)$

2 Solving DP recursions

- To improve our understanding of how this recursive representation works, let's solve the DP we just wrote for the knapsack problem
- We solve the DP backwards:
  - start with the boundary conditions in stage T
  - compute values of the cost-to-go function $f_t(n)$ in stages $T-1, T-2, \ldots, 3, 2$
  - …until we reach the desired cost-to-go function value
- Stage 4 computations – boundary conditions:

$$f_4(n) = 0 \qquad \text{for} \quad n = 0, 1, \ldots, 8$$

- Stage 3 computations:

$$f_3(8) = \max\{\overbrace{f_4(8)}^{x_t=0},\ \overbrace{12 + f_4(4)}^{x_t=1}\} = \max\{0, 12\} = 12$$

$$f_3(7) = \max\{f_4(7),\ 12 + f_4(3)\} = \max\{0, 12\} = 12$$

$$f_3(6) = \max\{f_4(6),\ 12 + f_4(2)\} = \max\{0, 12\} = 12$$

$$f_3(5) = \max\{f_4(5),\ \underbrace{12 + f_4(1)}_{x_3=1}\} = \max\{0, 12\} = 12$$

$$f_3(4) = \max\{f_4(4),\ 12 + f_4(0)\} = \max\{0, 12\} = 12$$

$$f_3(3) = \max\{f_4(3)\} = \max\{0\} = 0$$

$$f_3(2) = \max\{f_4(2)\} = \max\{0\} = 0$$

$$f_3(1) = \max\{f_4(1)\} = \max\{0\} = 0$$

$$f_3(0) = \max\{f_4(0)\} = \max\{0\} = 0$$

- Stage 2 computations:

$f_2(8) =$	$\max\{ f_3(8),\ 7 + f_3(6)\} = \max\{12,\ 7+12\} = 19$
$f_2(7) =$	$\max\{ f_3(7),\ 7 + f_3(5)\} = \max\{12,\ 7+12\} = 19$
$f_2(6) =$	$\max\{ f_3(6),\ 7 + f_3(4)\} = \max\{12,\ 7+12\} = 19$
$f_2(5) =$	$\max\{ \underbrace{f_3(5)}_{x_2=0},\ 7 + f_3(3)\} = \max\{12,\ 7\} = 12$
$f_2(4) =$	$\max\{ f_3(4),\ 7 + f_3(2)\} = \max\{12,\ 7\} = 12$
$f_2(3) =$	$\max\{ f_3(3),\ 7 + f_3(1)\} = \max\{0,\ 7\} = 7$
$f_2(2) =$	$\max\{ f_3(2),\ 7 + f_3(0)\} = \max\{0,\ 7\} = 7$
$f_2(1) =$	$\max\{ f_3(1)\} = \max\{0\} = 0$
$f_2(0) =$	$\max\{ f_3(0)\} = \max\{0\} = 0$

- Stage 1 computations – desired cost-to-go function:

$$f_1(8) = \max\{ f_2(8),\ \underbrace{11 + f_2(5)}_{x_1=1}\} = \max\{19,\ 11+12\} = 23$$

- Maximum value of theft: $f_1(8) = 23$

- Metals to take to achieve this maximum value:

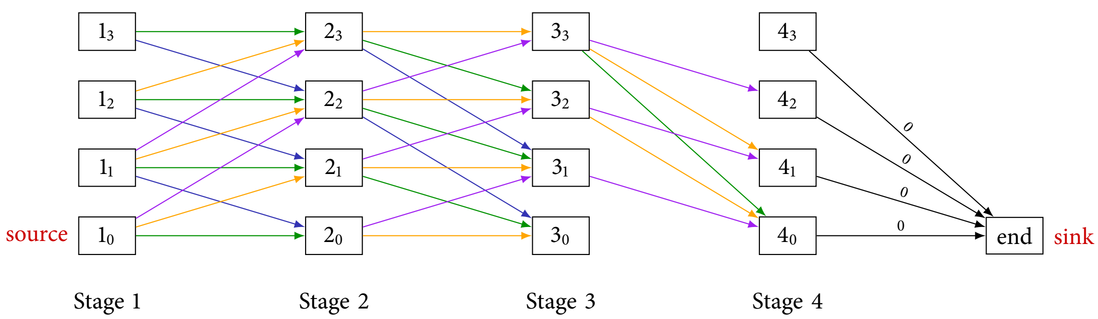
$x_1 = 1,\ x_2 = 0,\ x_3 = 1 \Rightarrow$ Take metals 1 and 3

Example 2. The Dijkstra Brewing Company is planning production of its new limited run beer, Primal Pilsner. The company must supply 1 batch next month, then 2 and 4 in successive months. Each month in which the company produces the beer requires a factory setup cost of \$5,000. Each batch of beer costs \$2,000 to produce. Batches can be held in inventory at a cost of \$1,000 per batch per month. Capacity limitations allow a maximum of 3 batches to be produced during each month. In addition, the size of the company's warehouse restricts the ending inventory for each month to at most 3 batches. The company has no initial inventory.

The company wants to find a production plan that will meet all demands on time and minimizes its total production and holding costs over the next 3 months. Formulate this problem as a dynamic program by giving its recursive representation. Solve the dynamic program.

Formulating the DP

- Recall that in Lesson 9, we formulated this problem as a dynamic program with the following shortest path representation:
  - Stage t represents the beginning of month t ($t = 1, 2, 3$) or the end of the decision-making process ($t = 4$).
  - Node t_n represents having n batches in inventory at stage t ($n = 0, 1, 2, 3$).



Month	Production amount	Edge		Edge length
1	0	$(1_n, 2_{n-1})$	for $n = 1, 2, 3$	$1(n-1)$
1	1	$(1_n, 2_n)$	for $n = 0, 1, 2, 3, 4$	$5 + 2(1) + 1(n)$
1	2	$(1_n, 2_{n+1})$	for $n = 0, 1, 2$	$5 + 2(2) + 1(n+1)$
1	3	$(1_n, 2_{n+2})$	for $n = 0, 1$	$5 + 2(3) + 1(n+2)$
2	0	$(2_n, 3_{n-2})$	for $n = 2, 3$	$1(n-2)$
2	1	$(2_n, 3_{n-1})$	for $n = 1, 2, 3$	$5 + 2(1) + 1(n-1)$
2	2	$(2_n, 3_n)$	for $n = 0, 1, 2, 3$	$5 + 2(2) + 1(n)$
2	3	$(2_n, 3_{n+1})$	for $n = 0, 1, 2$	$5 + 2(3) + 1(n+1)$
3	0	not possible		
3	1	$(3_n, 4_{n-3})$	for $n = 3$	$5 + 2(1) + 1(n-3)$
3	2	$(3_n, 4_{n-2})$	for $n = 2, 3$	$5 + 2(2) + 1(n-2)$
3	3	$(3_n, 4_{n-1})$	for $n = 1, 2, 3$	$5 + 2(3) + 1(n-1)$

- Let d_t = number of batches required in month t, for $t = 1, 2, 3$
- Stages:

$$\text{stage } t \longleftrightarrow \begin{cases} \text{beginning of month } t & \text{if } t = 1,2,3 \\ \text{end of process} & \text{if } t = 4 \end{cases}$$

- States:

$$\text{state } n \longleftrightarrow n \text{ batches in inventory} \quad \text{for } n = 0,1,2,3$$

- Allowable decisions x_t at stage t and state n:

$\underline{t = 1,2,3}$: Define x_t = # batches to produce in month t

x_t must satisfy: $x_t \in \{0, 1, 2, 3\}$ ← production capacity

$$0 \leq \underbrace{n + x_t - d_t}_{\text{new inventory}} \leq 3$$ ← inventory capacity

$\underline{t = 4}$: no decisions

- Reward of decision x_t at stage t and state n:

Let
$$I(x_t) = \begin{cases} 1 & \text{if } x_t > 0 \\ 0 & \text{o/w} \end{cases}$$

Reward:
$$5\, I(x_t) + 2 x_t + 1(n + x_t - d_t)$$
for $t = 1,2,3$; $n = 0,1,2,3$

- Reward-to go function $f_t(n)$ at stage t and state n:

$f_t(n)$ = minimum cost of meeting demand starting at month t with initial inventory of n batches

for $t = 1, \dots, 4$; $n = 0, \dots, 3$

- Boundary conditions:

$$f_4(n) = 0 \qquad \text{for } n = 0,1,2,3$$

- Recursion:

$$f_t(n) = \min_{\substack{x_t \in \{0,1,2,3\} \\ 0 \leq n + x_t - d_t \leq 3}} \left\{ 5\, I(x_t) + 2 x_t + 1(n + x_t - d_t) + f_{t+1}(n + x_t - d_t) \right\}$$

for $t = 1,2,3$; $n = 0,1,2,3$

- Desired reward-to-go function value:

$$f_1(0)$$

$$f_t(n) = \min_{\substack{x_t \in \{0,1,2,3\} \\ 0 \le n + x_t - d_t \le 3}} \left\{ 5 I(x_t) + 2x_t + 1(n + x_t - d_t) + f_{t+1}(n + x_t - d_t) \right\}$$
for $t = 1,2,3$; $n = 0,1,2,3$

Solving the DP

- Stage 4 computations – boundary conditions:

$$f_4(n) = 0 \quad \text{for } n = 0,1,2,3$$

- Stage 3 computations:

$(t=3)$

$x_3 \in \{0,1,2,3\}$
$0 \le 3 + x_3 - 4 \le 3$

$f_3(3) =$ $(n=3)$ $\min\{\underbrace{5 + 2(1) + 1(0) + f_4(0)}_{x_3=1}^{7},\ \underbrace{5 + 2(2) + 1(1) + f_4(1)}_{x_3=2}^{10},\ \underbrace{5 + 2(3) + 1(2) + f_4(2)}_{x_3=3}^{13}\} = 7$

$x_3 \in \{0,1,2,3\}$
$0 \le 2 + x_3 - 4 \le 3$
$\Rightarrow x_3 \in \{2,3\}$

$f_3(2) =$ $\min\{\underbrace{5 + 2(2) + 1(0) + f_4(0)}_{x_3=2}^{9},\ \underbrace{5 + 2(3) + 1(1) + f_4(1)}_{x_3=3}^{12}\} = 9$

$x_3 \in \{3\}$

$f_3(1) =$ $\min\{\underbrace{5 + 2(3) + 1(0) + f_4(0)}_{x_3=3}^{11}\} = 11$

$x_3 \in \{0,1,2,3\}$
$0 \le 0 + x_3 - 4 \le 3$
$\Rightarrow$ no allowable decisions

$f_3(0) =$ $+\infty$

- Stage 2 computations:

$0 \le n + x_2 - d_2 \le 3$
$0 \le 3 + x_2 - 2 \le 3$
$0 \le x_2 + 1 \le 3$

$f_2(3) =$ $\min\{\underbrace{1(1) + f_3(1)}_{x_2=0}^{12},\ \underbrace{5 + 2(1) + 1(2) + f_3(2)}_{x_2=1}^{18},\ \underbrace{5 + 2(2) + 1(3) + f_3(3)}_{x_2=2}^{19}\} = 12$

$f_2(2) =$ $\min\{\underbrace{1(0) + f_3(0)}_{x_2=0}^{+\infty},\ \underbrace{5 + 2(1) + 1(1) + f_3(1)}_{x_2=1}^{19},$
$\underbrace{5 + 2(2) + 1(2) + f_3(2)}_{x_2=2}^{20},\ \underbrace{5 + 2(3) + 1(3) + f_3(3)}_{x_2=3}^{21}\} = 19$

$f_2(1) =$ $\min\{\underbrace{5 + 2(1) + 1(0) + f_3(0)}_{x_2=1}^{+\infty},\ \underbrace{5 + 2(2) + 1(1) + f_3(1)}_{x_2=2}^{21},\ \underbrace{5 + 2(3) + 1(2) + f_3(2)}_{x_2=3}^{22}\} = 21$

$f_2(0) =$ $\min\{\underbrace{5 + 2(2) + 1(0) + f_3(0)}_{x_2=2}^{+\infty},\ \underbrace{5 + 2(3) + 1(1) + f_3(1)}_{x_2=3}^{23}\} = 23$

- Stage 1 computations – desired cost-to-go function:

$x_1 \in \{1,2,3\}$

$f_1(0) = \min\{\underbrace{5 + 2(1) + 1(0) + f_2(0)}_{x_1=1}^{30},\ \underbrace{5 + 2(2) + 1(0) + f_2(1)}_{x_1=2}^{31},\ \underbrace{5 + 2(3) + 1(2) + f_2(2)}_{x_1=3}^{32}\}$
$= 30$

- Minimum total production and holding cost: $f_1(0) = 30$

- Production amounts that achieve this minimum value:

$x_1 = 1,\quad x_2 = 3,\quad x_3 = 3 \quad \Longrightarrow$ Produce 1 batch in month 1
3 batches in months 2 and 3