Celso Carneiro Ribeiro

Introdução aos Modelos e Métodos de Otimização em Pesquisa Operacional

Parte IV – Programação Dinâmica

Operador Nacional do Sistema Elétrico

2004

Parte IV – Programação Dinâmica

- Origens
- Motivação
- Programação dinâmica
- Aplicações
- Programação dinâmica estocástica
- Aplicação

Origens

- Trabalhos de Bellman (anos 50)
- Aplicação a problemas de decisões seqüenciais:
  - Estoques
  - Substituição de equipamentos
  - Operação energética
  - Jogos
- Princípio da otimalidade: "toda subtrajetória da trajetória ótima é ótima com relação a suas extremidades inicial e final".

Bibliografia

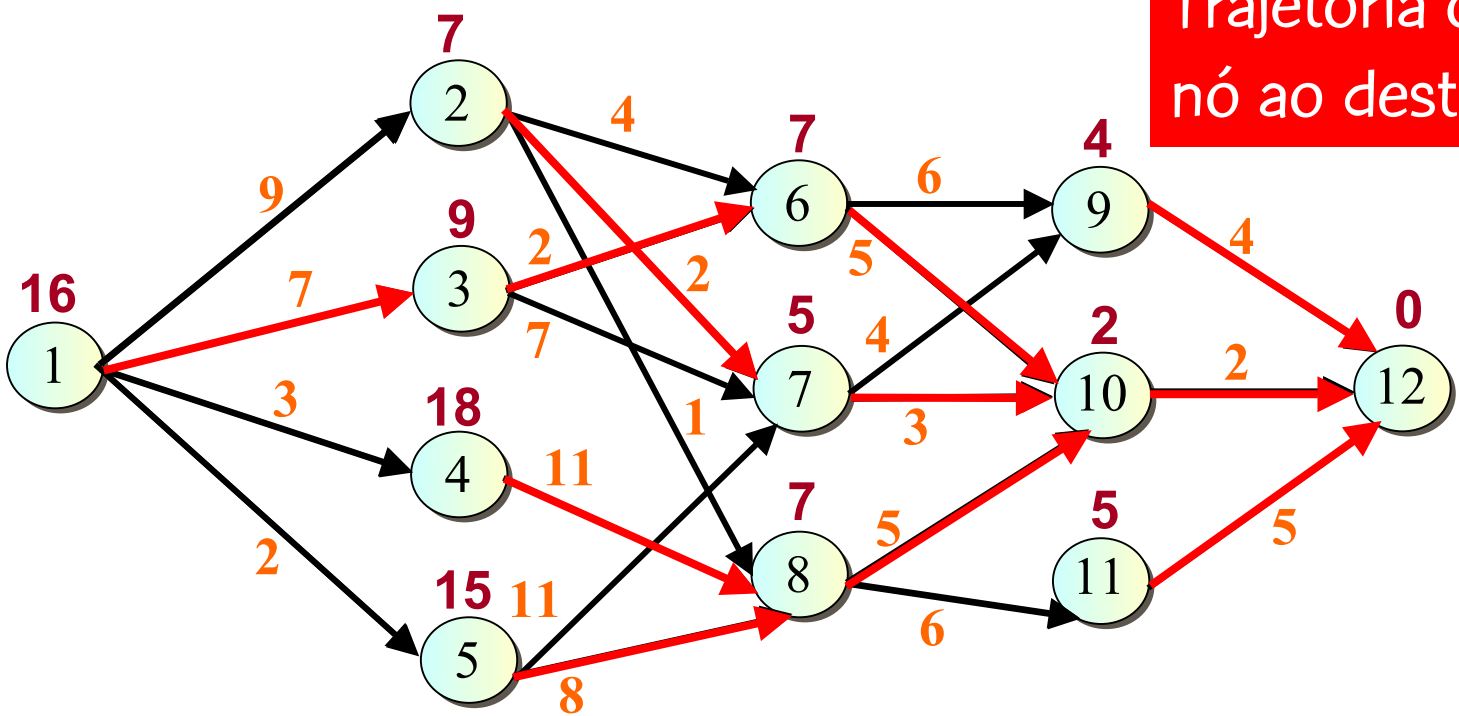
- R. Bellman, "Dynamic Programming", 2003.
- R. Bellman, "Applied Dynamic Programming", 1962.
- G. Nemhauser, "Introduction to Dynamic Programming", 1966.
- D. Bertsekas, "Dynamic Programming and Optimal Control", 2001.
- E. Denardo, "Dynamic Programming: Models and Applications", 2003.
- M. Sniedovich, "Dynamic Programming", 2004.

...

Motivação

- Problema: determinar o caminho mais curto de 1 a 12 no grafo abaixo

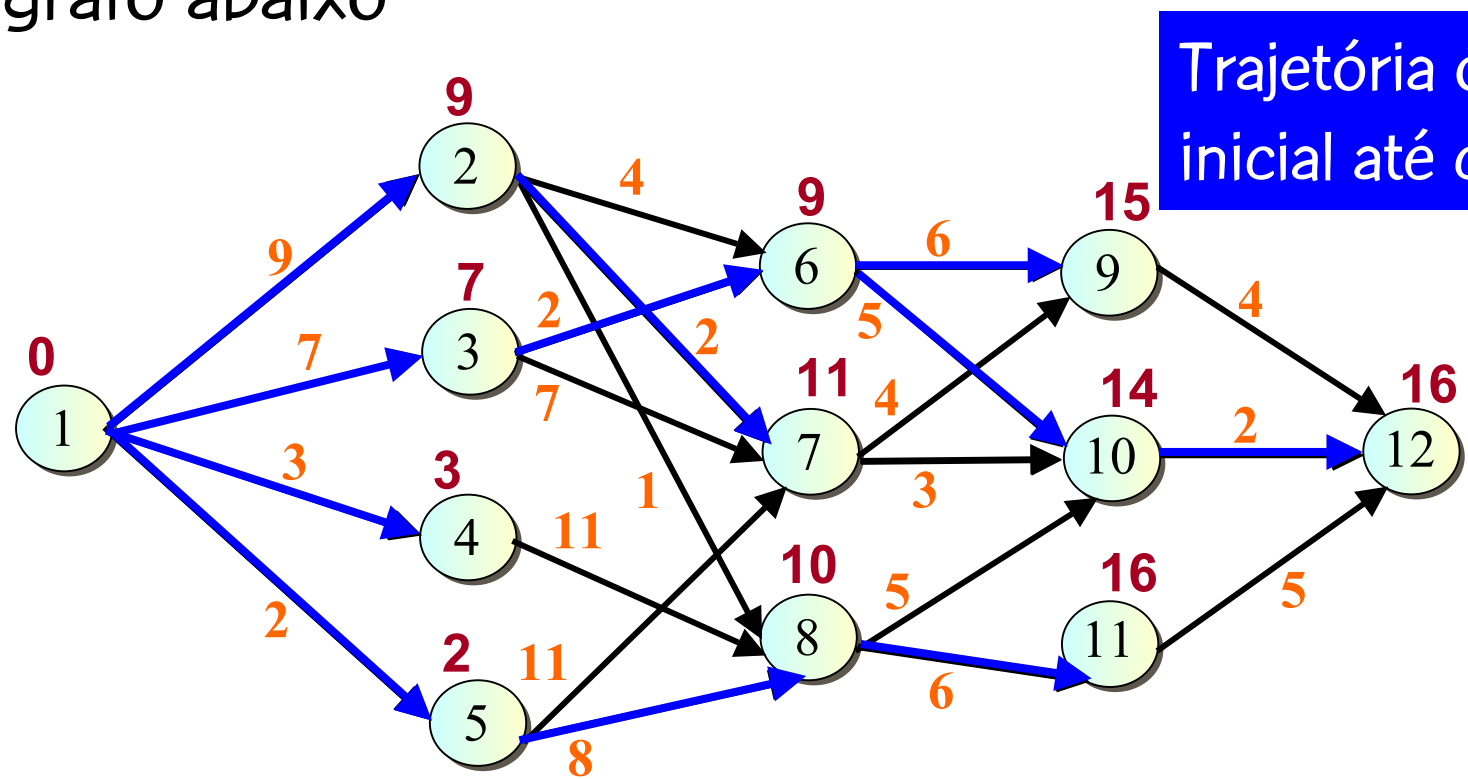
Trajetória ótima de cada nó ao destino final



- Procedimento backwards

Motivação

- Problema: determinar o caminho mais curto de 1 a 12 no grafo abaixo

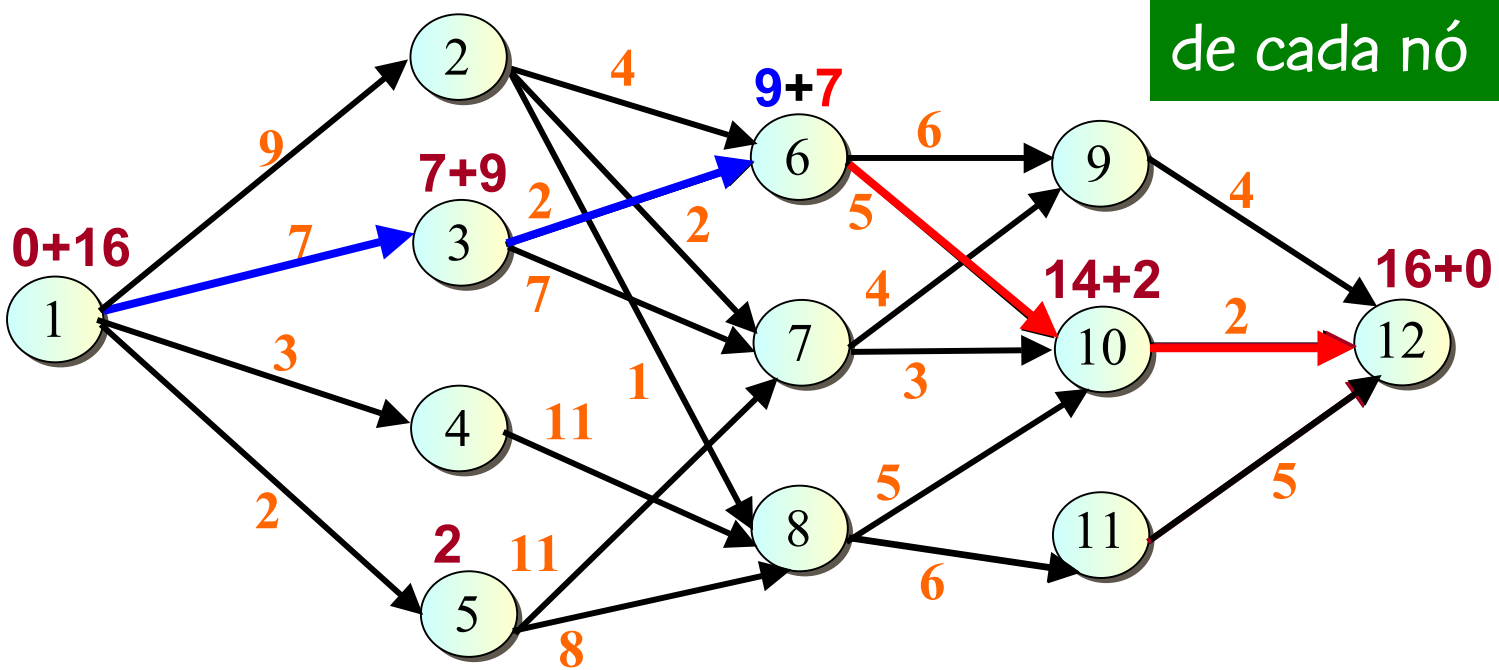


- Procedimento forward

Motivação

- Problema: determinar o caminho mais curto de 1 a 12 no grafo abaixo

Trajetória ótima através de cada nó

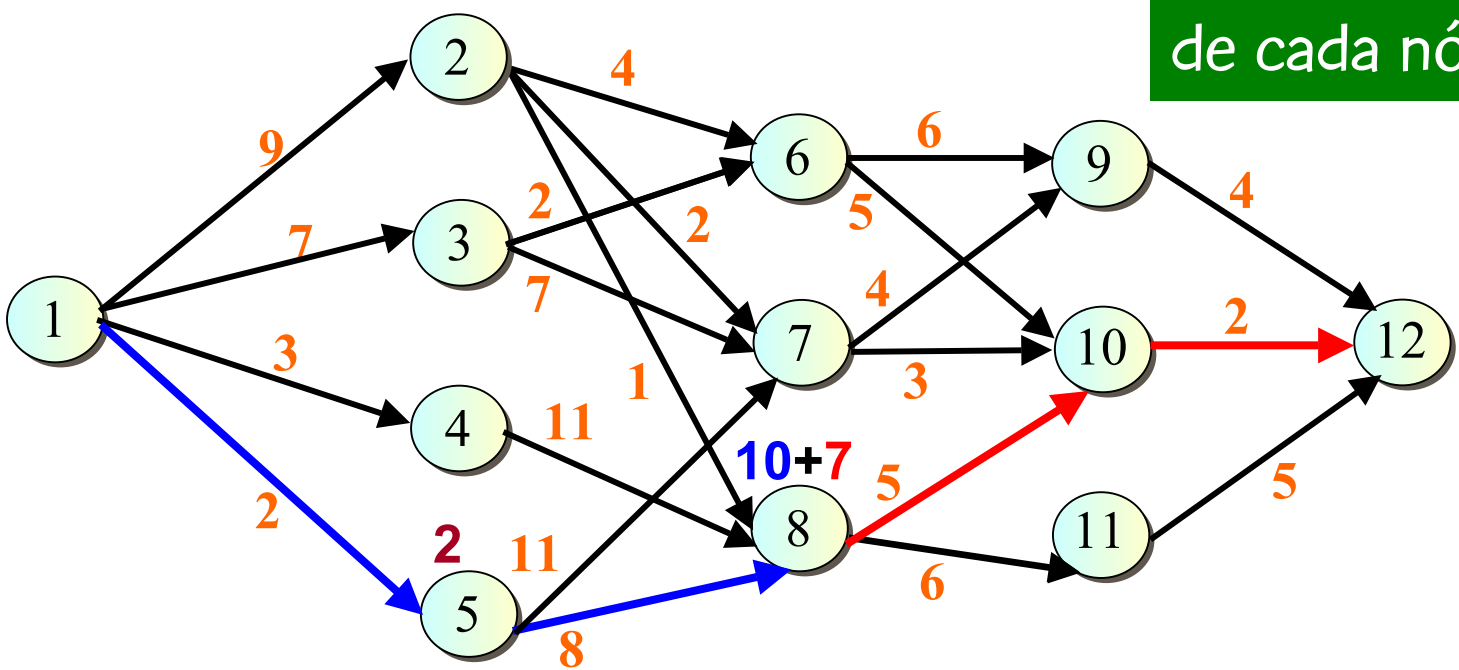


- Combinação das trajetórias ótimas

Motivação

- Problema: determinar o caminho mais curto de 1 a 12 no grafo abaixo

Trajetória ótima através de cada nó



- Combinação das trajetórias ótimas

Programação dinâmica

- Aplicação a problemas de decisões seqüenciais: cada decisão aplicada a um estado em determinado estágio leva a um estado do estágio imediatamente seguinte.
- Princípio da otimalidade: uma seqüência ótima de decisões tem a propriedade de que quaisquer que sejam o estado e a decisão inicial, as decisões remanescentes constituem uma seqüência ótima de decisões com relação ao estado decorrente da primeira decisão.
- Alternativamente: "toda subtrajetória da trajetória ótima é ótima com relação a suas extremidades inicial e final".

Programação dinâmica

- Método exato para resolver problemas de progamação inteira que envolvem apenas decisões seqüenciais, nos quais cada nova decisão depende apenas do estado do sistema, mas não das decisões anteriores (isto é, da forma como este estado foi atingido).
- Principais conceitos envolvidos:
  - estágios (etapas)
  - estados
  - decisões
  - função critério a ser otimizada

Programação dinâmica

- Roteiro de aplicação:
  - Identificar um modelo de decisões seqüenciais através de seus estágios.
  - Assegurar-se de que cada solução viável (ou trajetória) pode ser vista como uma seqüência de decisões tomadas a cada estágio, de modo tal que seu custo seja igual à soma dos custos das decisões individuais.
  - Definir o conceito de estado como a resultante de todas as decisões relevantes tomadas no passado (caso forward).

    (alternativamente, definir o conceito de estado como a resultante de todas as decisões relevantes tomadas no futuro no caso backwards)

Programação dinâmica

- Roteiro de aplicação (continuação):
  - Determinar as transições de estado possíveis.
  - Atribuir o custo de cada transição de estado à decisão correspondente.
  - Escrever uma recursão que defina o custo ótimo do estado inicial até o estado final.
- Exemplo de aplicação: problema da mochila

Problema da mochila

- Caso (4): os itens não podem ser fracionados e no máximo uma unidade de cada item pode ser selecionada

- Problema de programação inteira
- As variáveis inteiras binárias (0-1) representam a decisão de selecionar um objeto ou não.
- Solução não trivial!

$$\text{maximizar} \sum_{j=1}^{n} c_j x_j$$

$$\text{sujeito a:} \sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j \in \{0,1\}, \quad j = 1,\ldots,n$$

- Ordenar as variáveis pelo índice lucro/volume resolve o problema linear apenas, mas não o de programação inteira.

Problema da mochila

- Decomposição do problema em estágios
- Em vez de considerar uma solução $(x_1,x_2,\ldots,x_n)$ completa de uma só vez, visualizar o problema como se as decisões fossem tomadas para um item de cada vez.
- Após k decisões, terão sido determinados quais dos primeiros k itens devem ser selecionados e, conseqüentemente, terão sido determinados os valores das variáveis $x_1,x_2,\ldots,x_k$.
- Neste ponto, o valor acumulado é $\sum_{j=1}^{k} c_j x_j$ e o volume acumulado é $\sum_{j=1}^{k} a_j x_j$.

Problema da mochila

- Estágio:

  cada variável do problema
- Estado:

  volume total ocupado com as decisões já tomadas
- Decisão:

  selecionar ou não um item (isto é, fazer $x_k=1$)
- Custo da decisão de selecionar o item k:

  aumento de c_k unidades no lucro parcial acumulado
- Efeito da decisão de selecionar o item k:

  aumento do volume ocupado (estado) em a_k unidades

Problema da mochila

- Definição:
  $y_k(u)$ = lucro máximo que pode ser obtido com volume total igual a u e usando apenas itens do conjunto {1,...,k}
- Quanto vale $y_0(0)$?
  - $y_0(0) = 0$ (sem objetos selecionados, o peso e o lucro são nulos)
- Definir $y_k(u) = -\infty$ se é impossível obter um volume total igual a u apenas com itens dentre os do conjunto {1,...,k}.
- Quanto valem $y_0(u)$ e $y_k(0)$?
  - $y_0(u) = -\infty$ para $u > 0$ (impossível acumular peso sem itens)
  - $y_k(0) = 0$ para $k = 1,2,...,n$ (nenhum item selecionado)
- Calcular $y_k(u)$ para $k = 1,...,n$ e $u = 0,...b$, a partir de $y_0(0)$.

Problema da mochila

- $y_k(u)$ = lucro máximo que pode ser obtido com volume total igual a u e usando apenas itens do conjunto {1,...,k}
- Calcular $y_k(u)$ para k = 1,...,n e u = 0,...b:

$$y_k(u) = \begin{cases} 0, & u = 0; k = 0 \\ -\infty, & u = 1,...,b; k = 0 \\ \max\{y_{k-1}(u), y_{k-1}(u - a_k) + c_k\}, & u = 0,...,b; k = 1,...,n \end{cases}$$

- Interpretação: há duas alternativas para se obter $y_k(u)$, dependendo do item k ser selecionado ou não
  - $y_k(u) = y_{k-1}(u)$, se o item k não é usado
  - $y_k(u) = y_{k-1}(u-a_k)+c_k$, se o item k é usado

Problema da mochila

$$y_k(u) = \begin{cases} 0, & u = 0; k = 0 \\ -\infty, & u = 1,\ldots,b; k = 0 \\ \max\{y_{k-1}(u), y_{k-1}(u - a_k) + c_k\}, & u = 0,\ldots,b; k = 1,\ldots,n \end{cases}$$

- Observar que o lucro associado ao estado resultante de uma decisão depende apenas do valor desta decisão e do estado atual, mas não depende da forma como este último foi atingido.

Problema da mochila

$$y_5(4) = \max 3x_1 + 2x_2 + 2x_3 + x_4 + 3x_5$$

$$\text{sujeito a}: 3x_1 + 3x_2 + x_3 + x_4 + 2x_5 = 4$$

$$x_j \in \{0,1\},\ j = 1,\ldots,5$$

Problema da mochila

$$\max 3x_1 + 2x_2 + 2x_3 + x_4 + 3x_5$$

$$\text{sujeito a}: 3x_1 + 3x_2 + x_3 + x_4 + 2x_5 \leq 4$$

$$x_j \in \{0,1\},\ j = 1,\ldots,5$$

$y_5(b) =$

$$\max 3x_1 + 2x_2 + 2x_3 + x_4 + 3x_5$$

$$\text{sujeito a}: 3x_1 + 3x_2 + x_3 + x_4 + 2x_5 = b$$

$$x_j \in \{0,1\},\ j = 1,\ldots,5$$

Valor ótimo = $\max_{b=0,\ldots,4} \{y_5(b)\}$

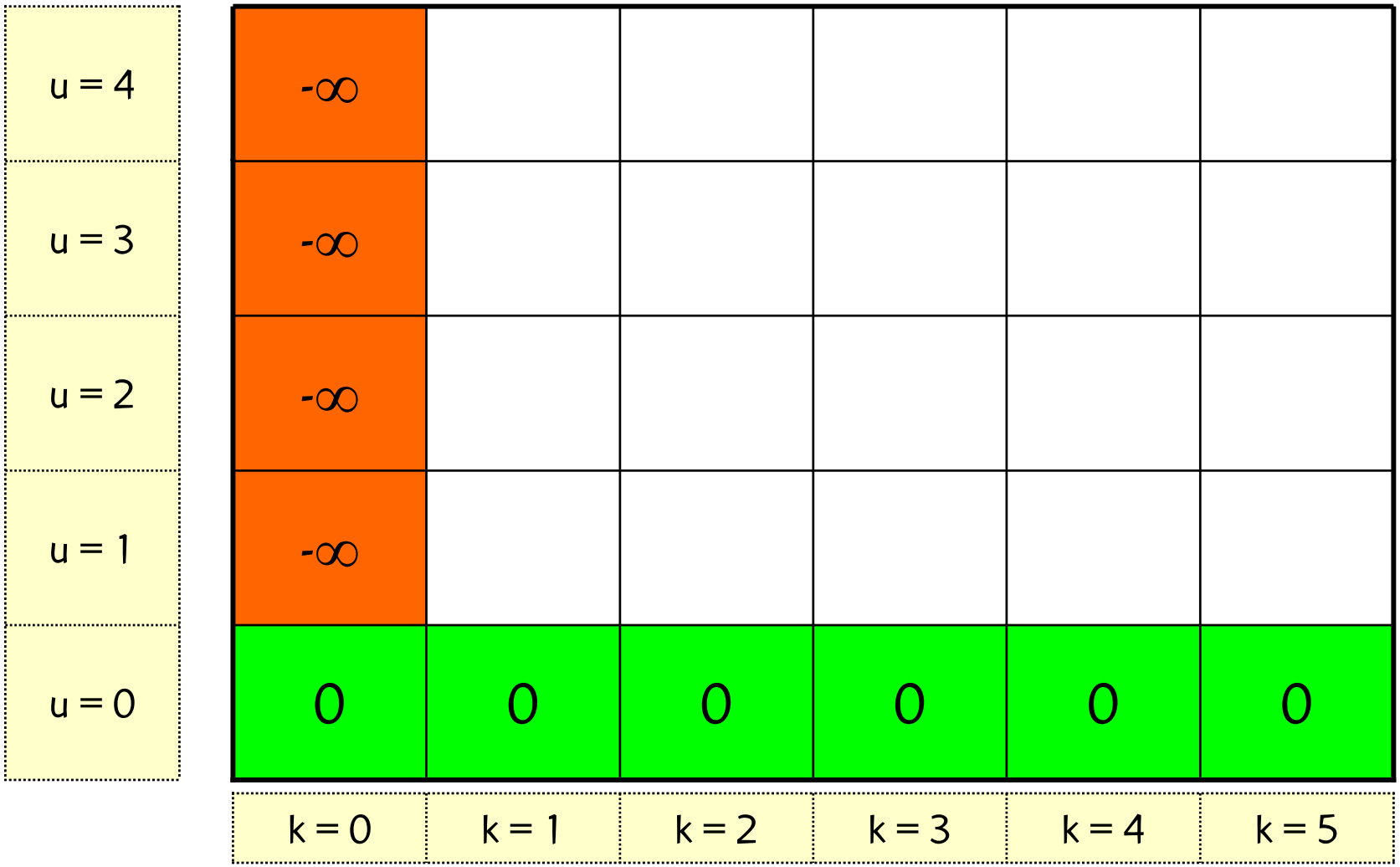
Problema da mochila

u = 4	$y_0(4)$					$y_5(4)$
u = 3	$y_0(3)$					$y_5(3)$
u = 2	$y_0(2)$					$y_5(2)$
u = 1	$y_0(1)$					$y_5(1)$
u = 0	$y_0(0)$	$y_1(0)$	$y_2(0)$	$y_3(0)$	$y_4(0)$	$y_5(0)$
	k = 0	k = 1	k = 2	k = 3	k = 4	k = 5

Problema da mochila

u = 4	$y_0(4)$					
u = 3	$y_0(3)$					
u = 2	$y_0(2)$					
u = 1	$y_0(1)$					
u = 0	0	0	0	0	0	0
	k = 0	k = 1	k = 2	k = 3	k = 4	k = 5

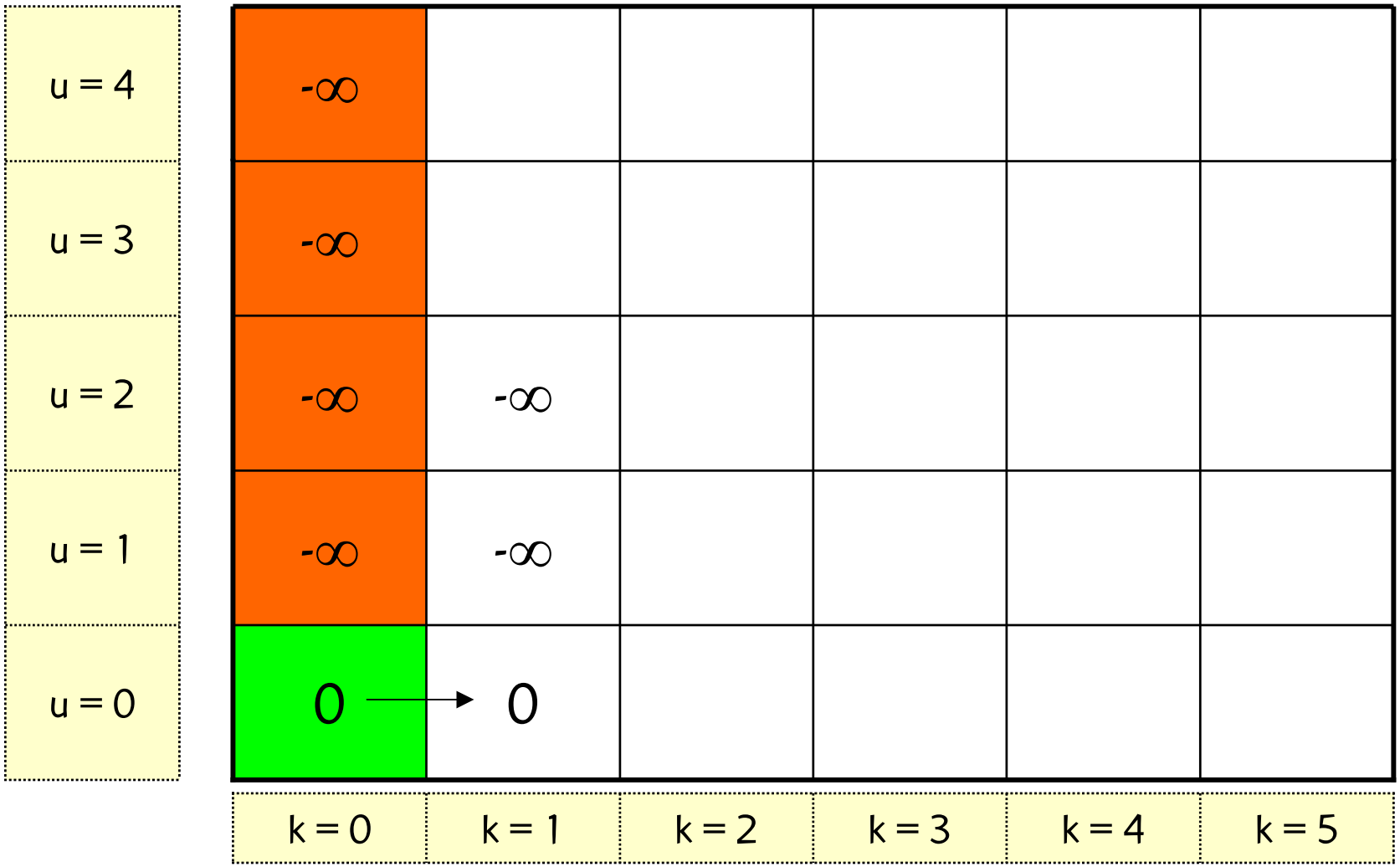
Problema da mochila



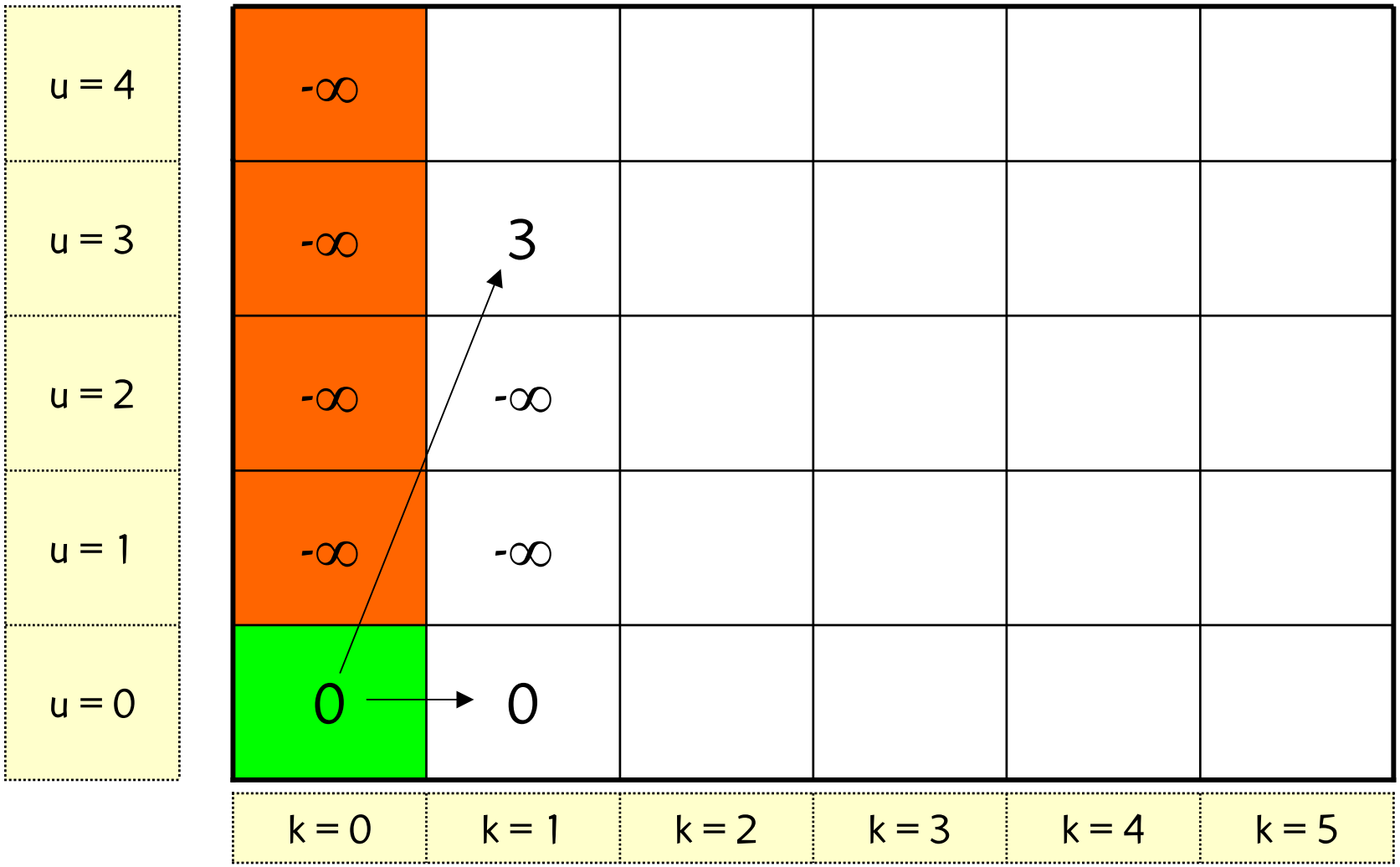
Problema da mochila

u = 4	-∞	?				
u = 3	-∞	?				
u = 2	-∞	?				
u = 1	-∞	?				
u = 0	0	?				
	k = 0	k = 1	k = 2	k = 3	k = 4	k = 5

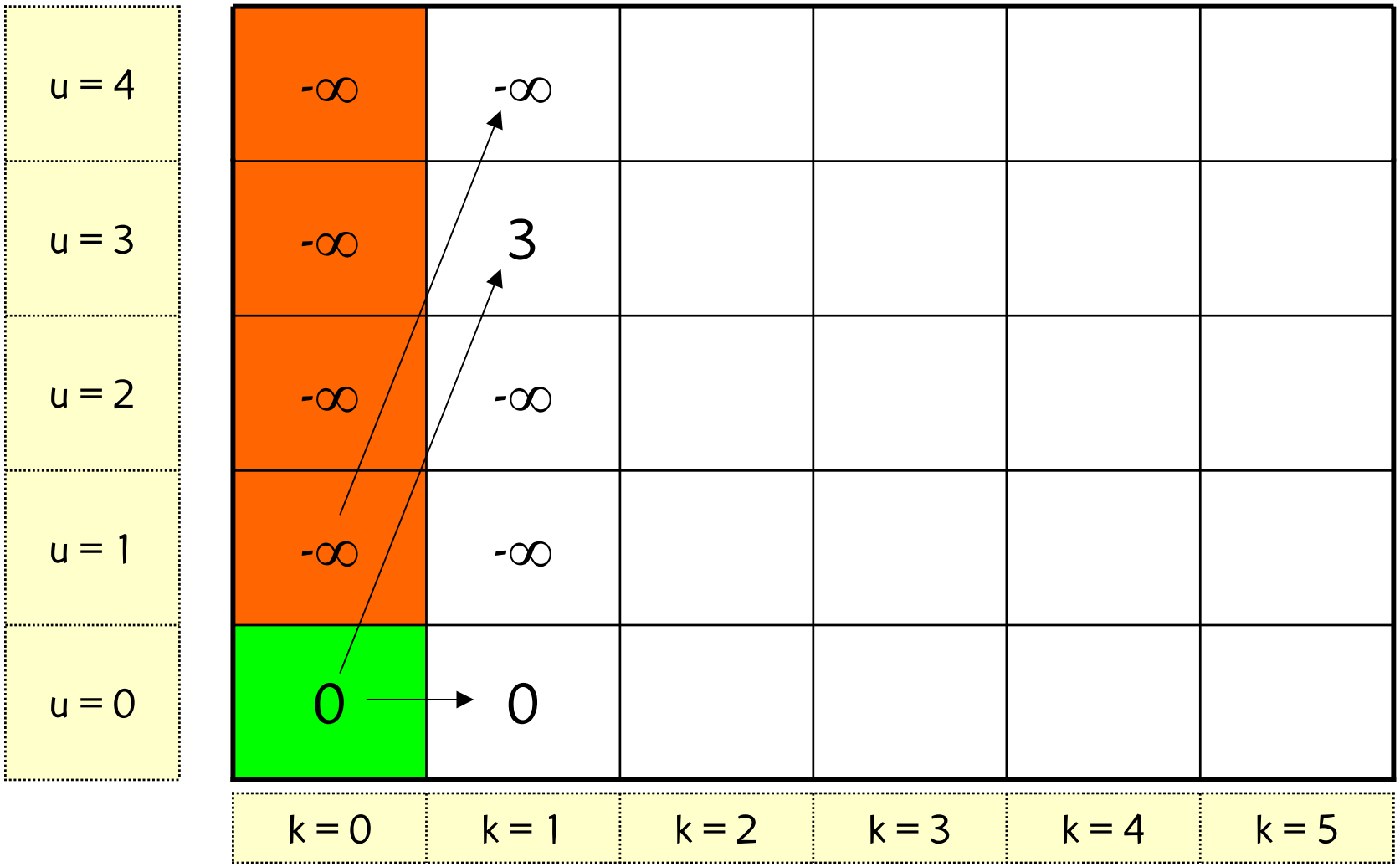
Problema da mochila



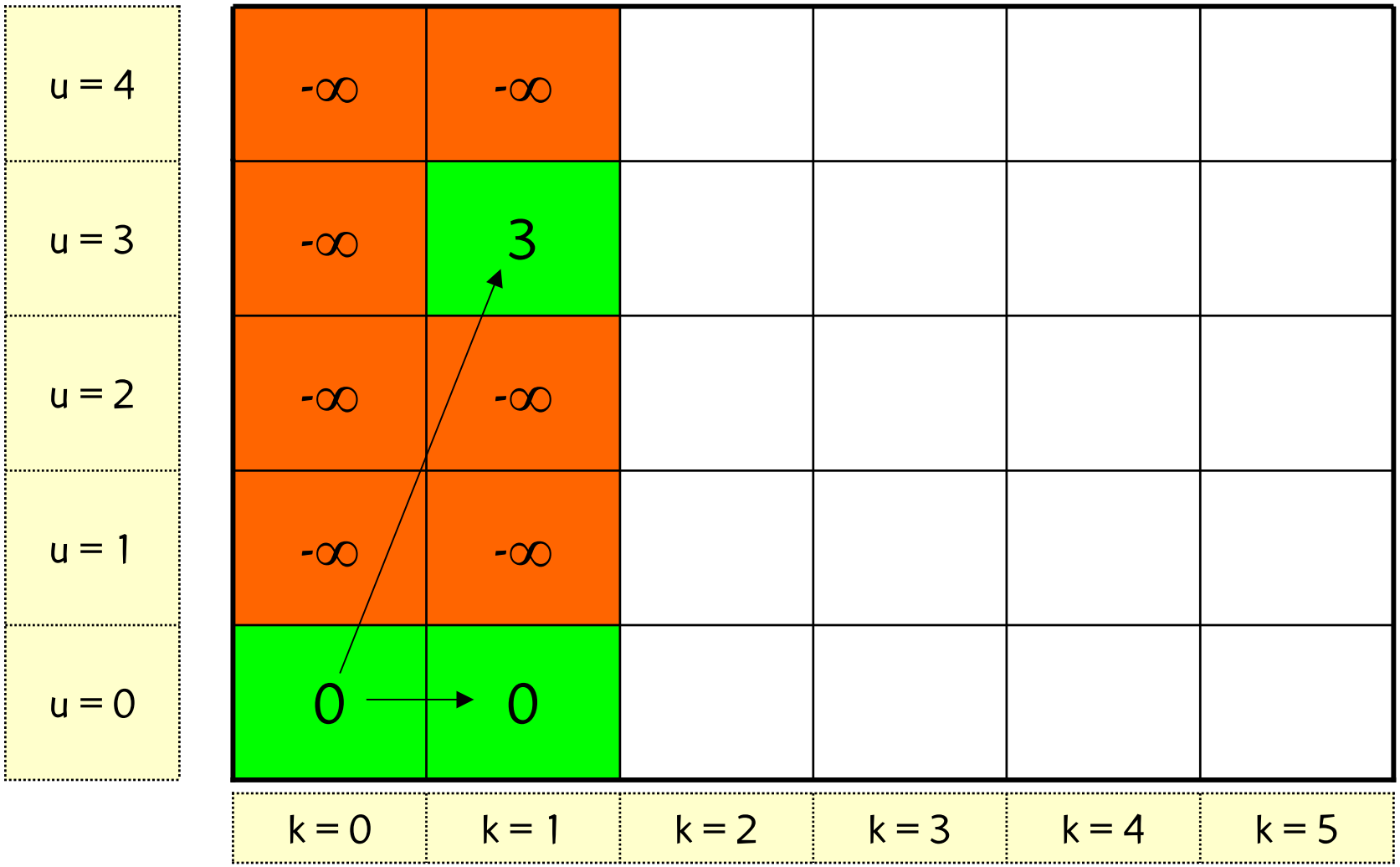
Problema da mochila



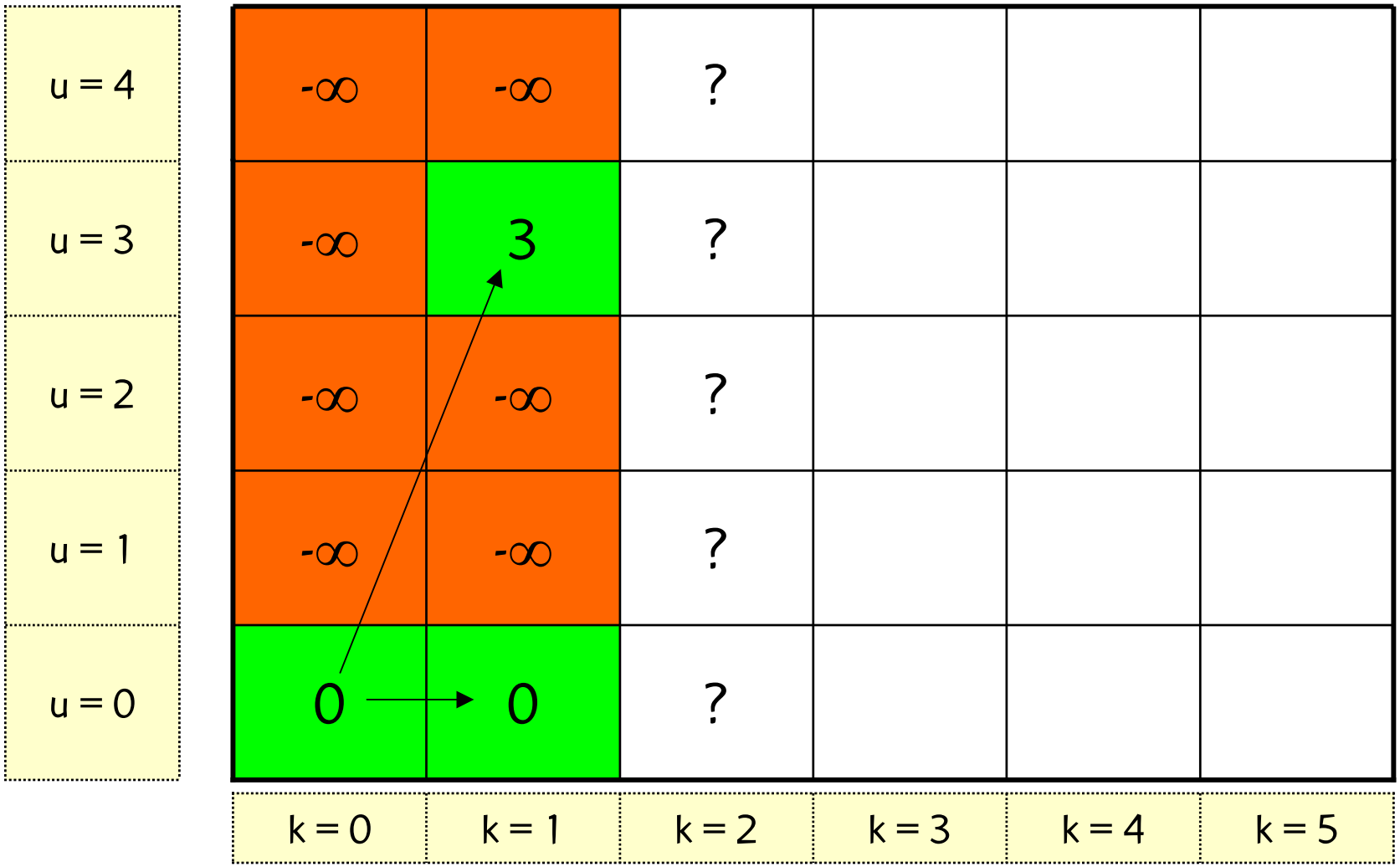
Problema da mochila



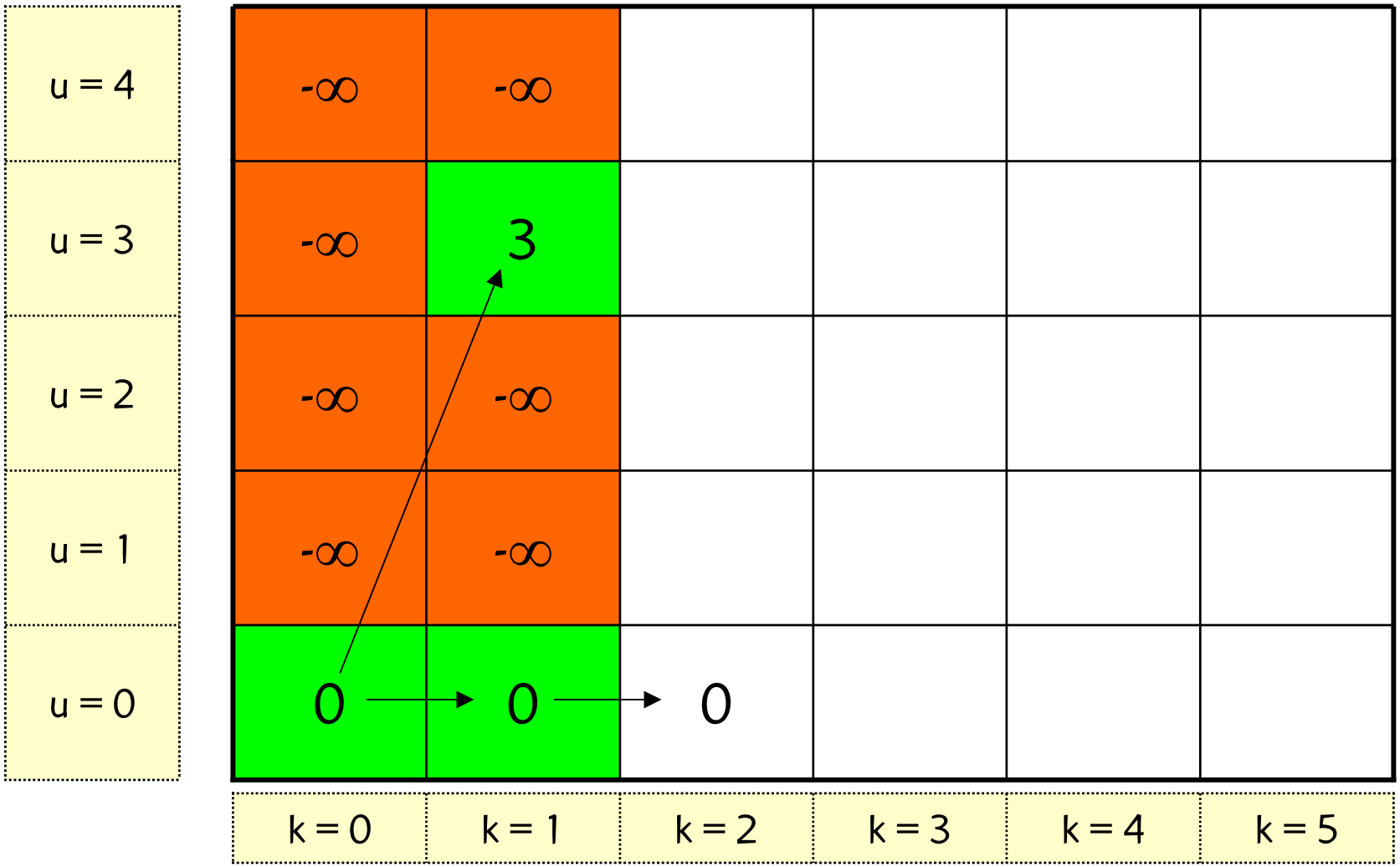
Problema da mochila



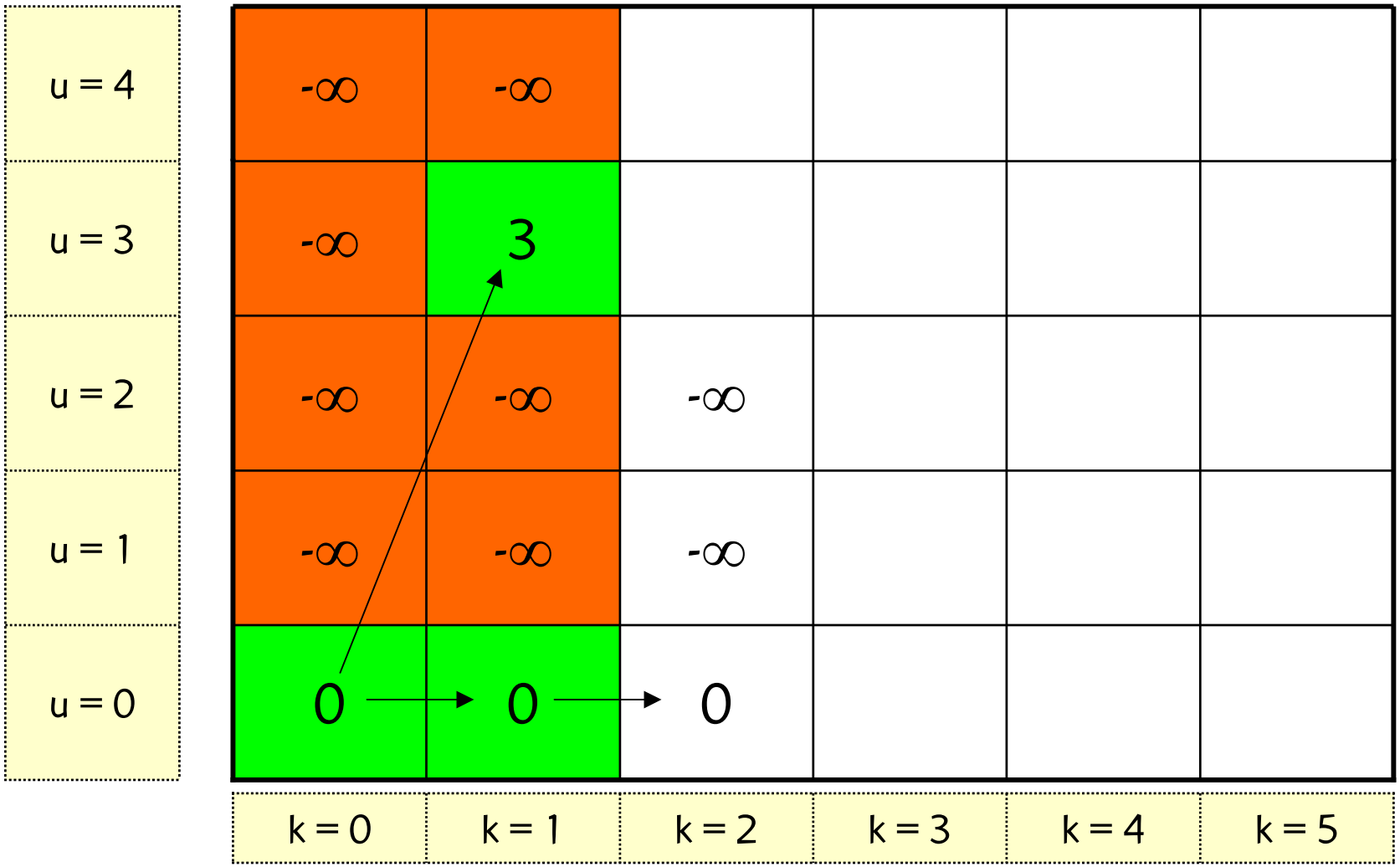
Problema da mochila



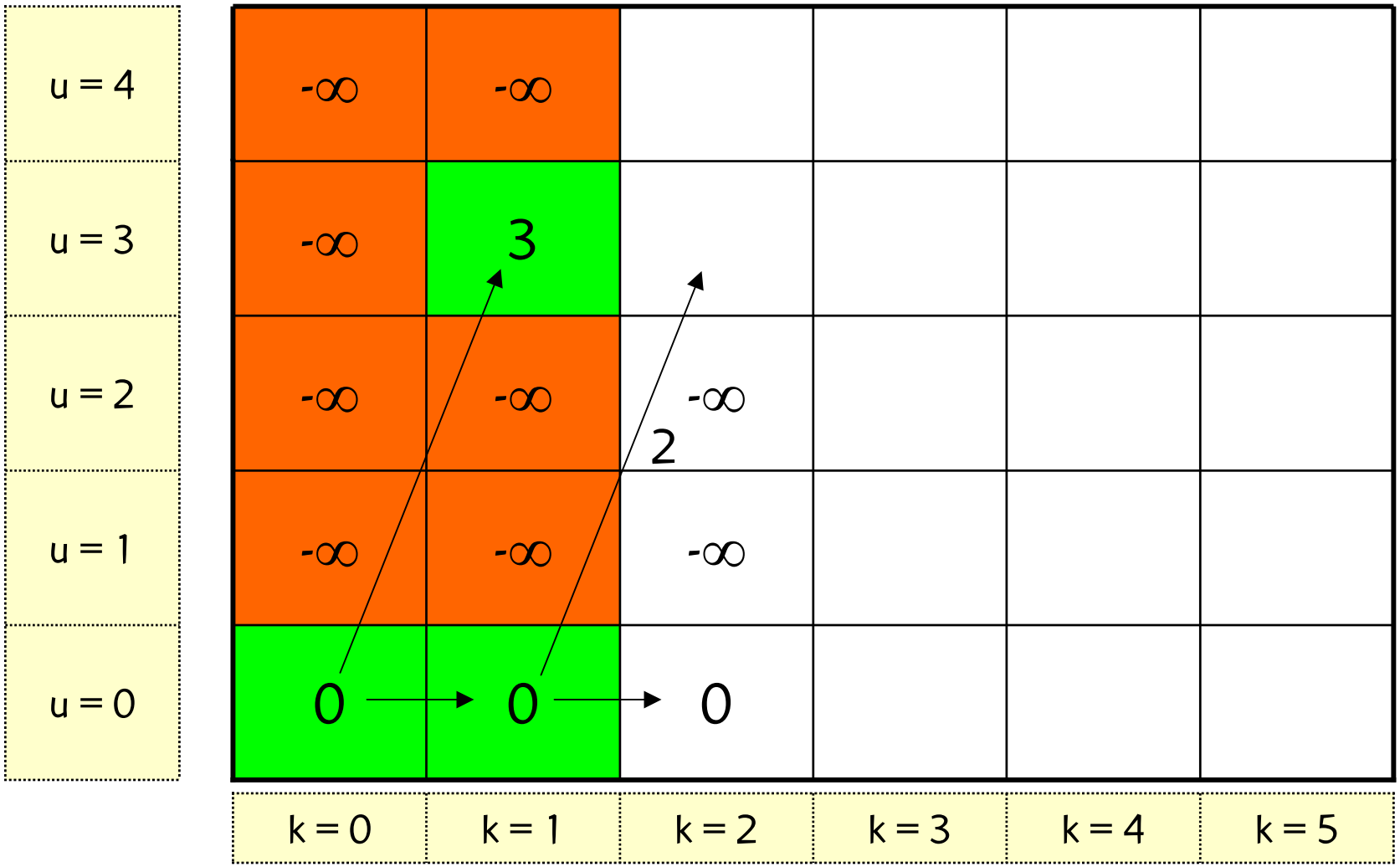
Problema da mochila



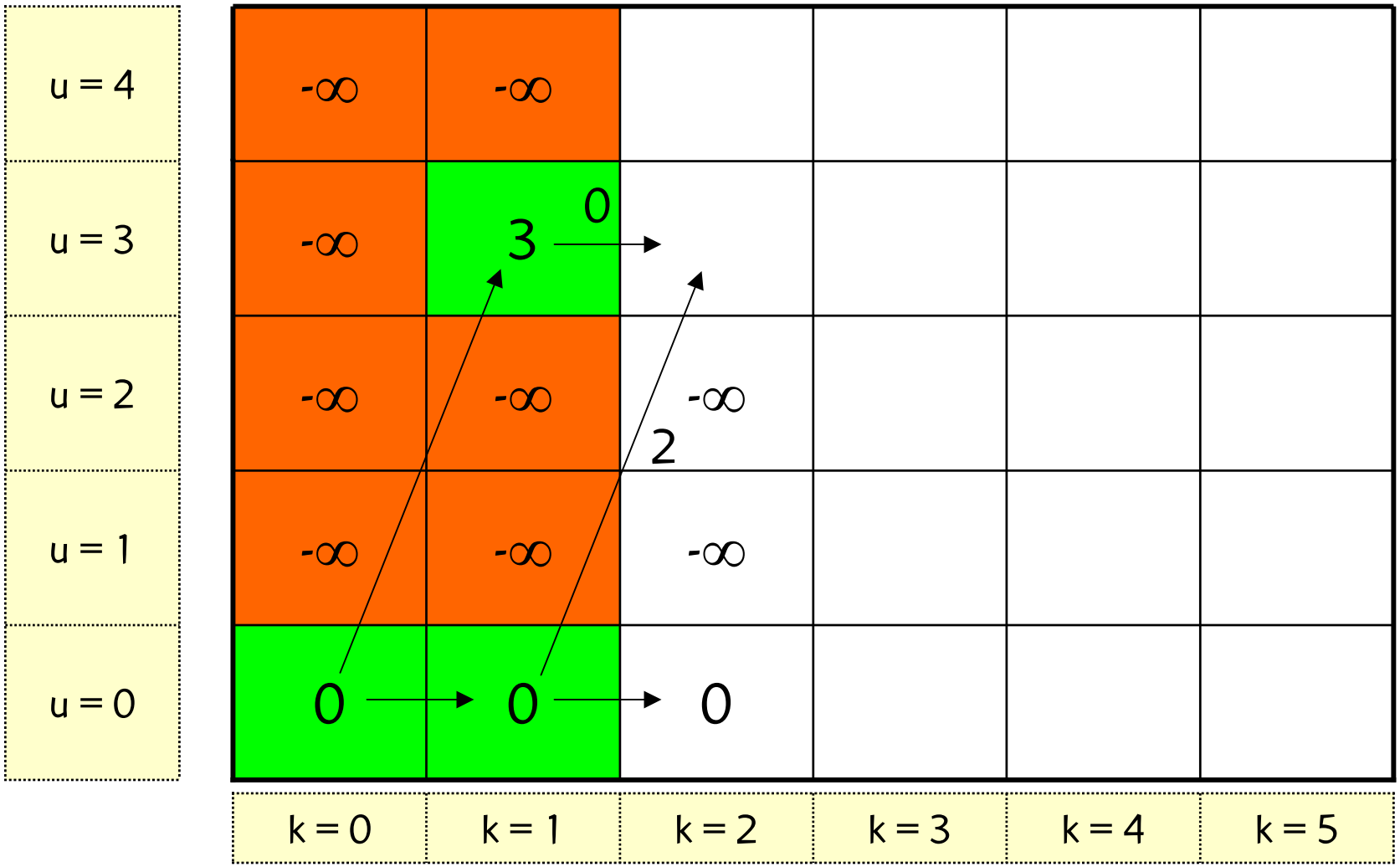
Problema da mochila



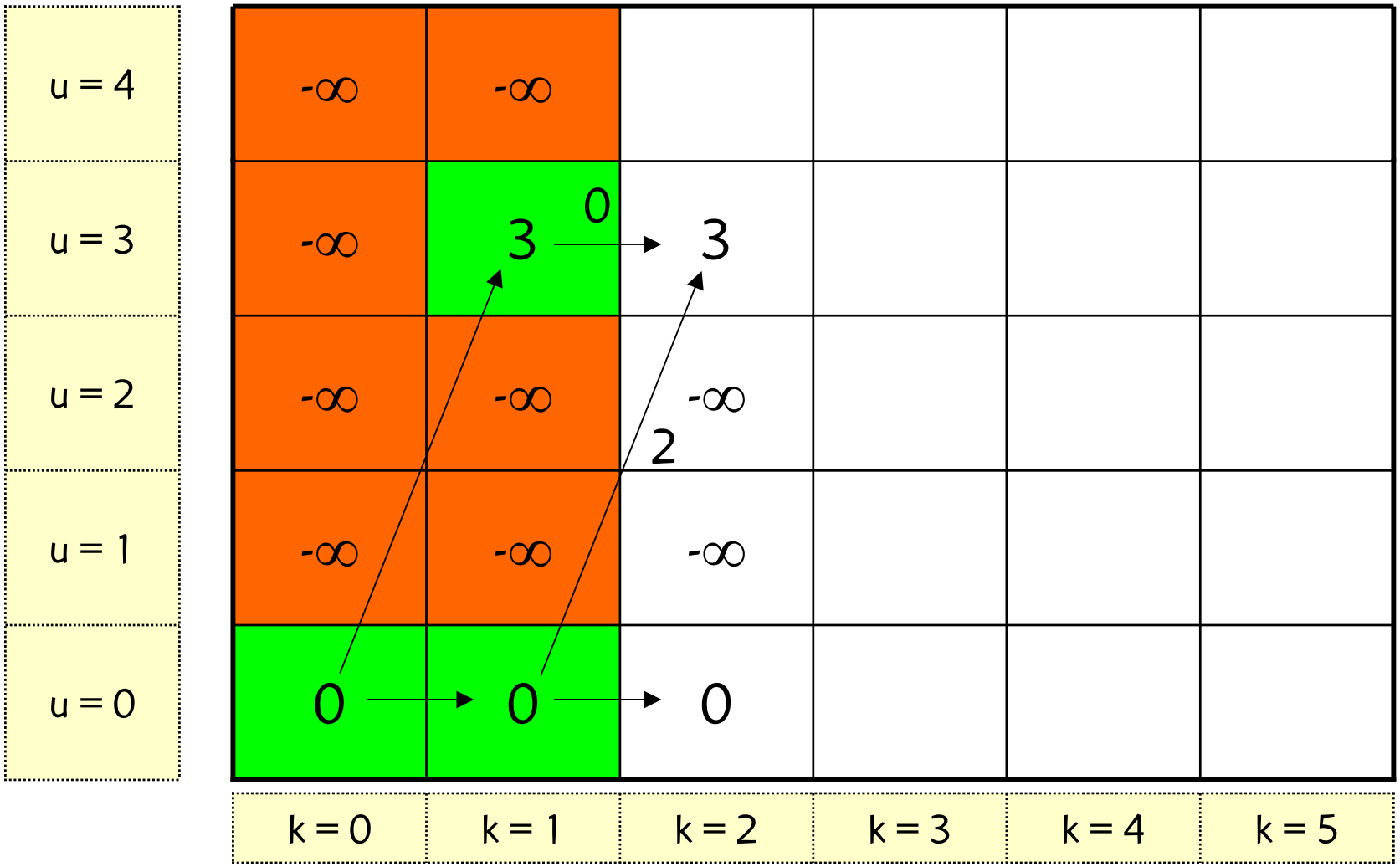
Problema da mochila



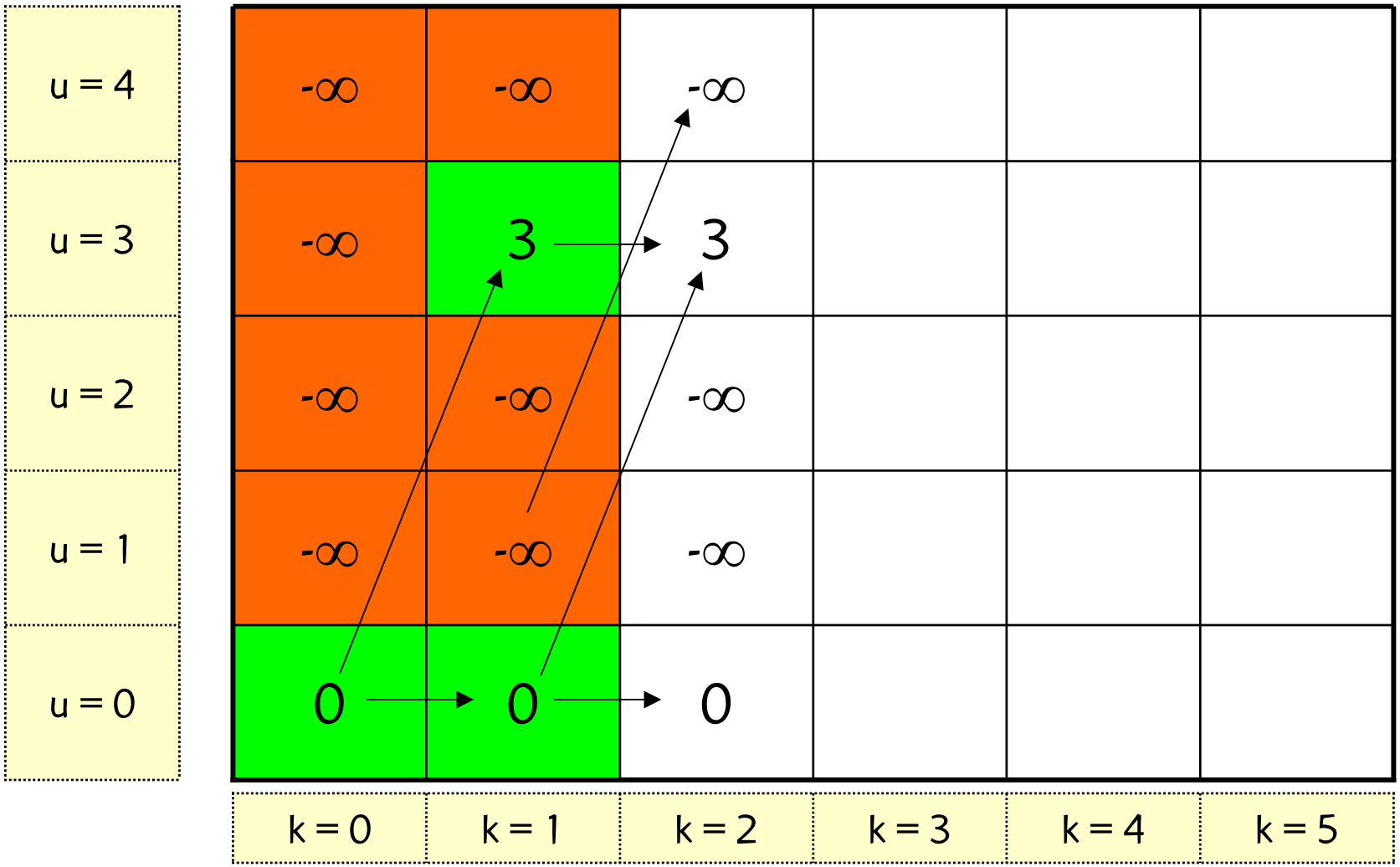
Problema da mochila



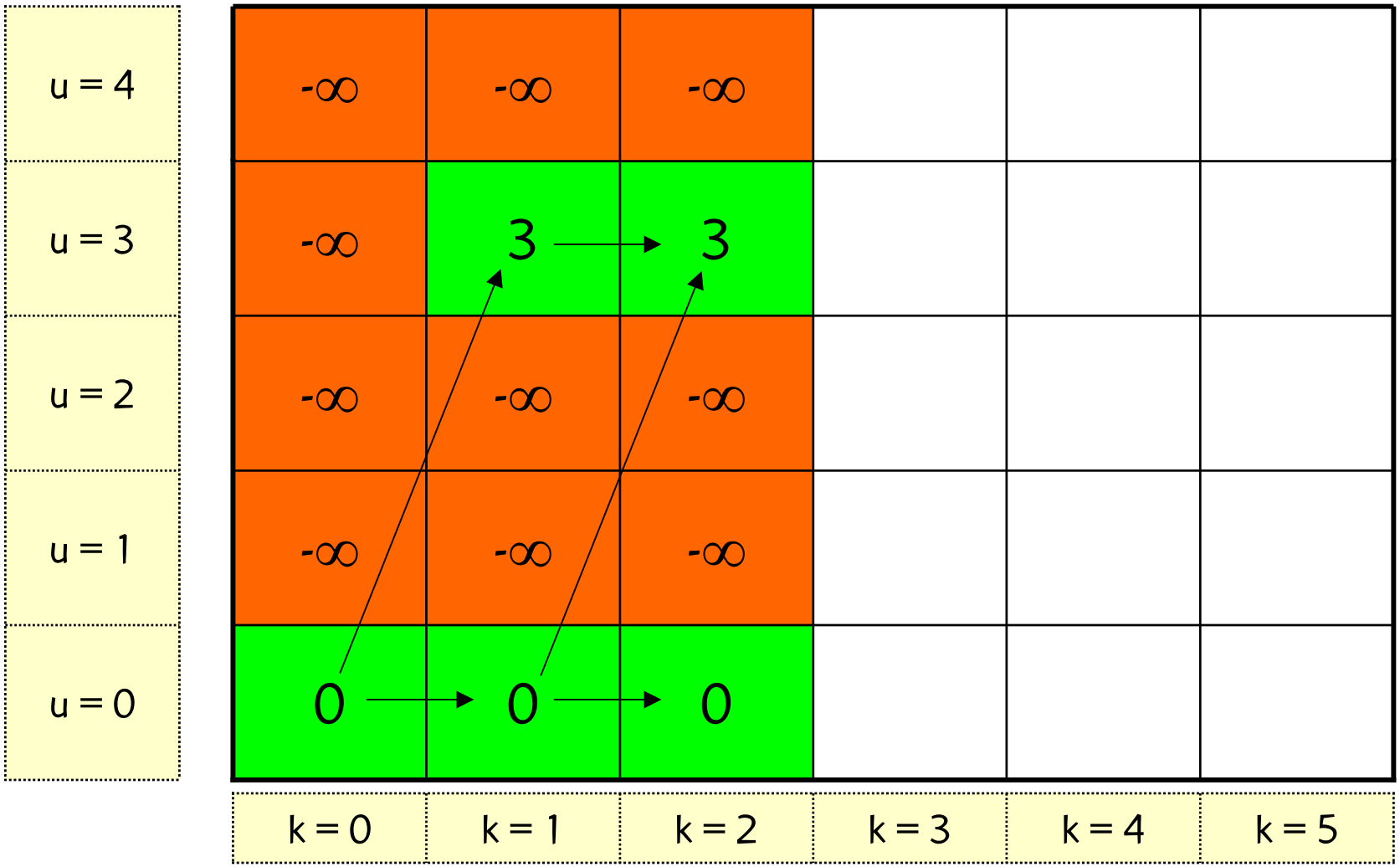
Problema da mochila



Problema da mochila

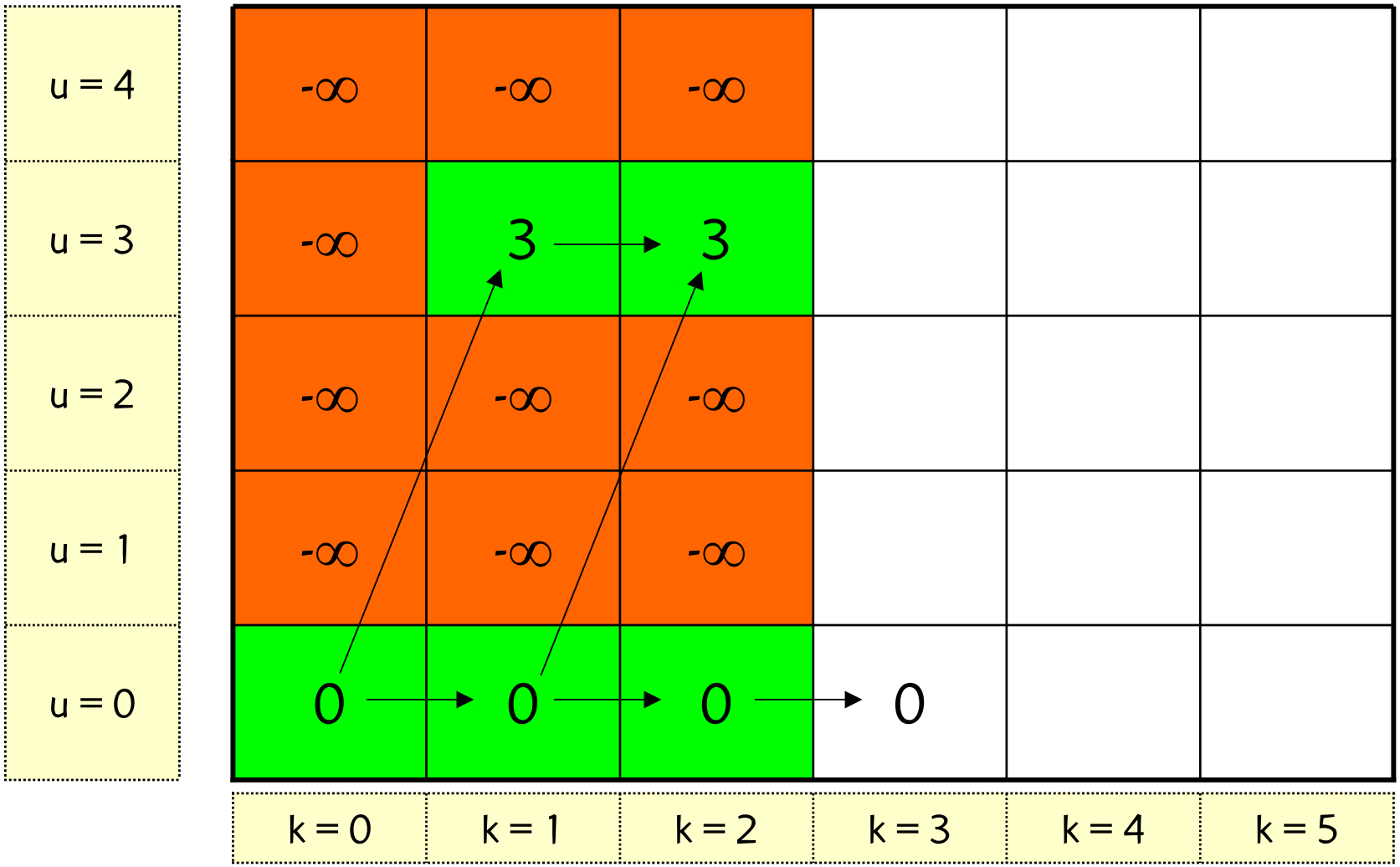


Problema da mochila

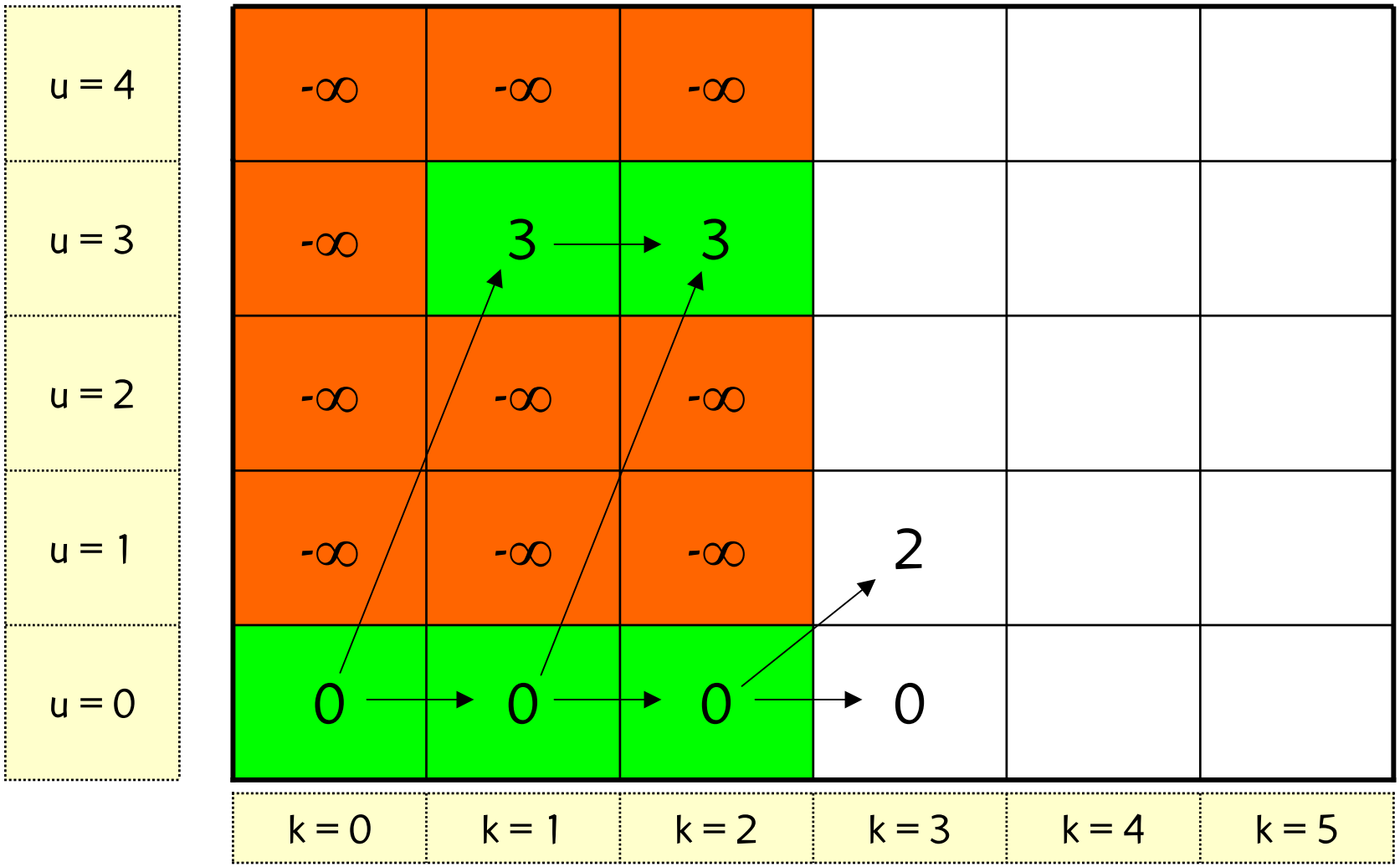


Problema da mochila

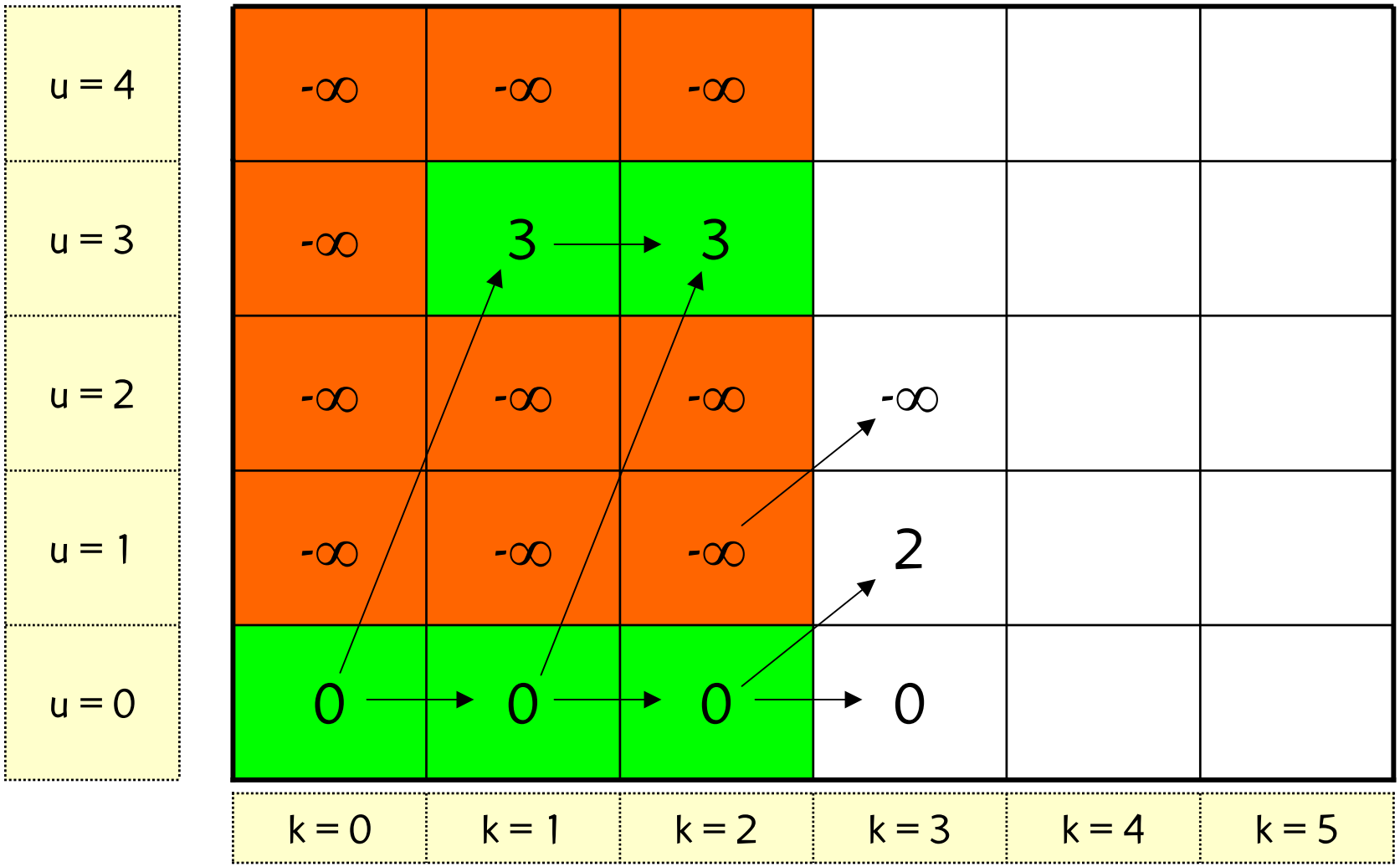
Problema da mochila



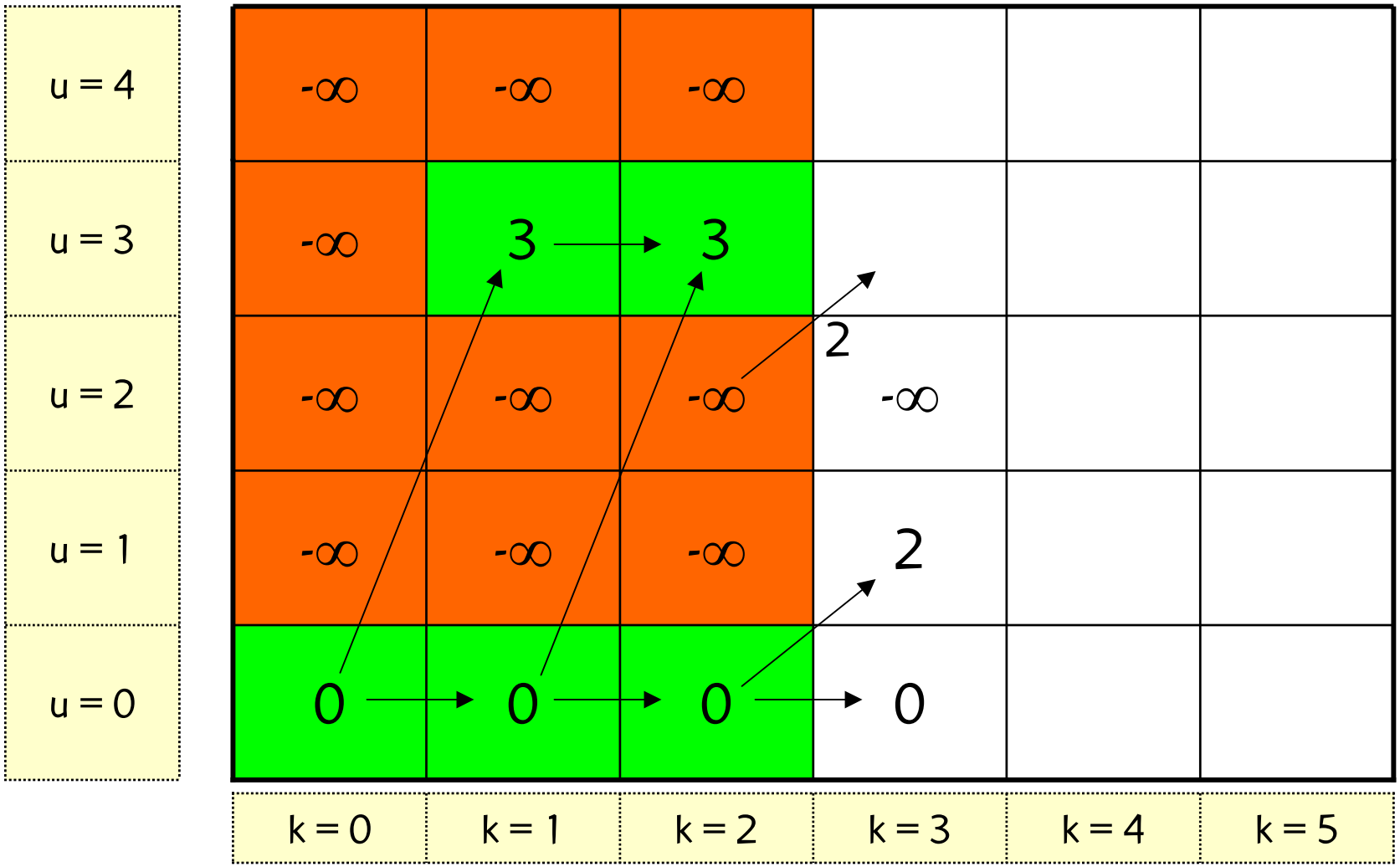
Problema da mochila



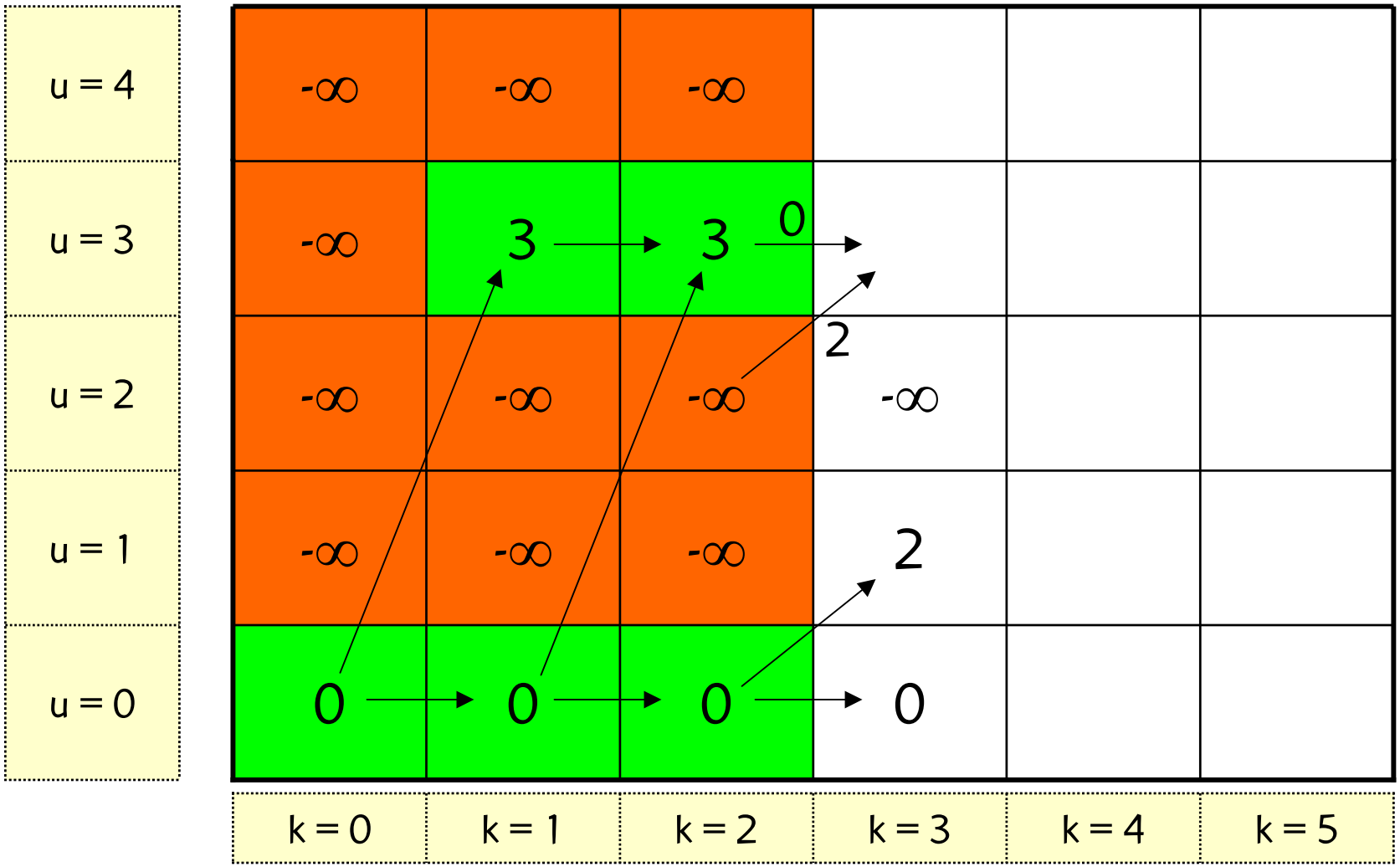
Problema da mochila



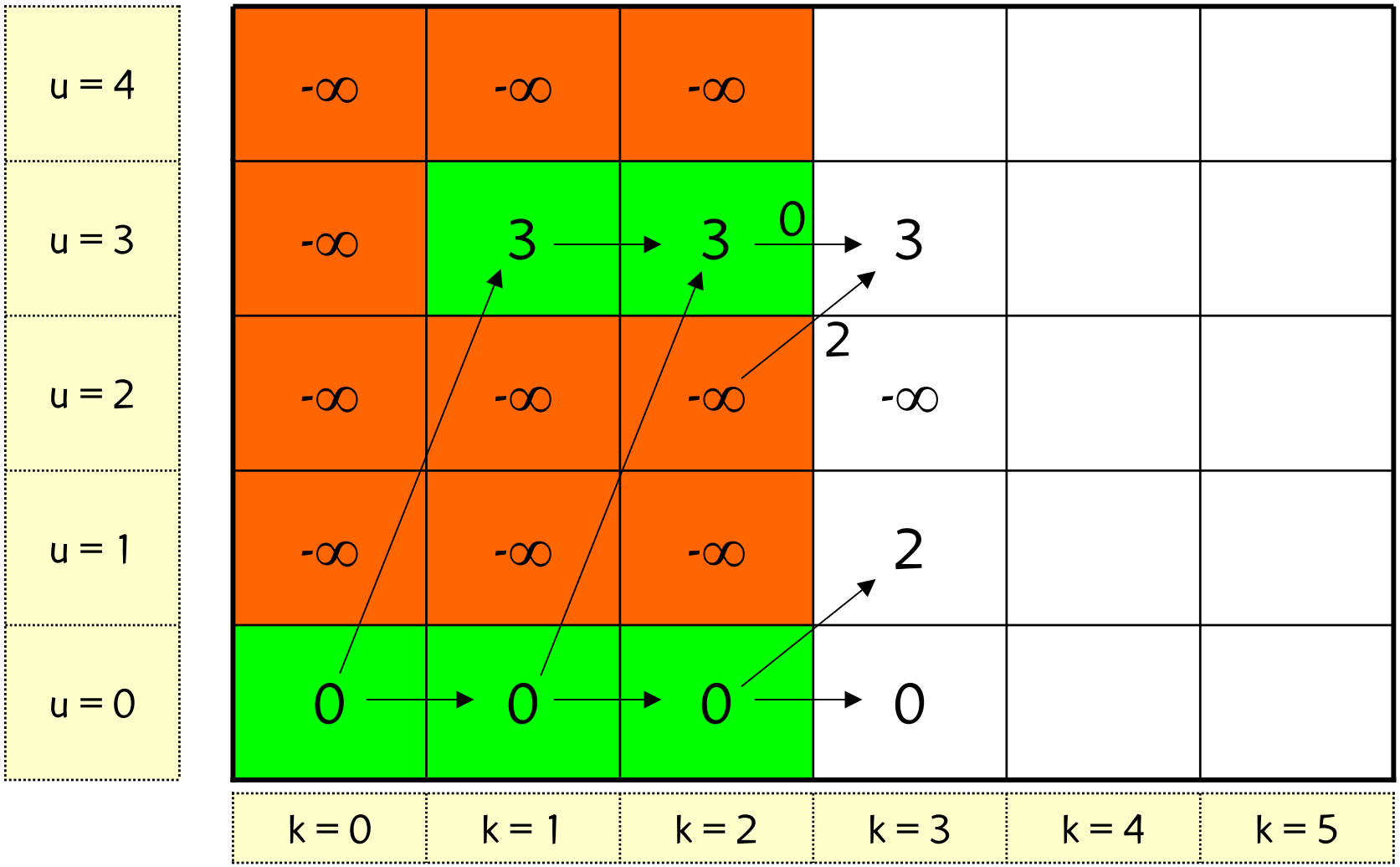
Problema da mochila



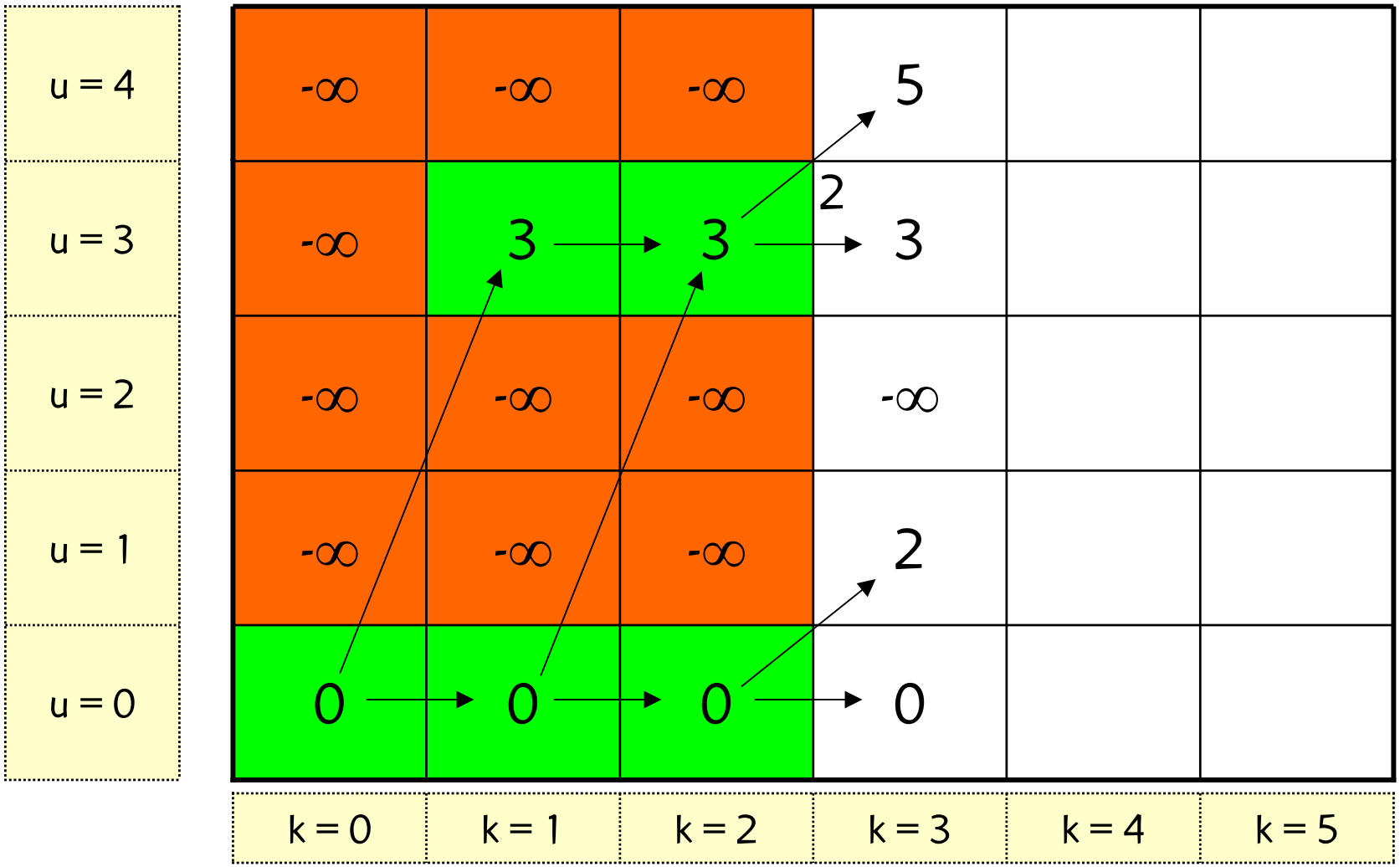
Problema da mochila



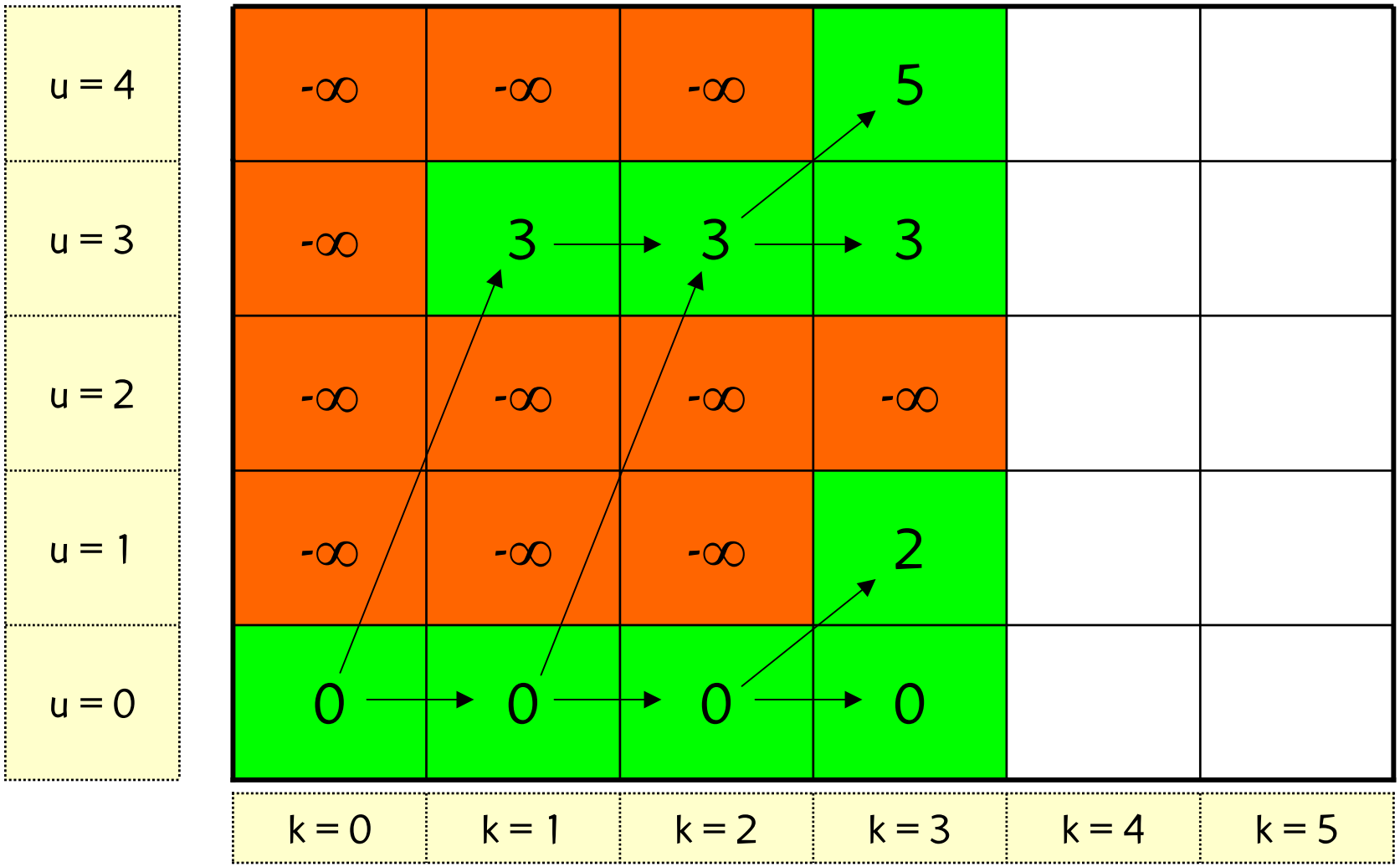
Problema da mochila



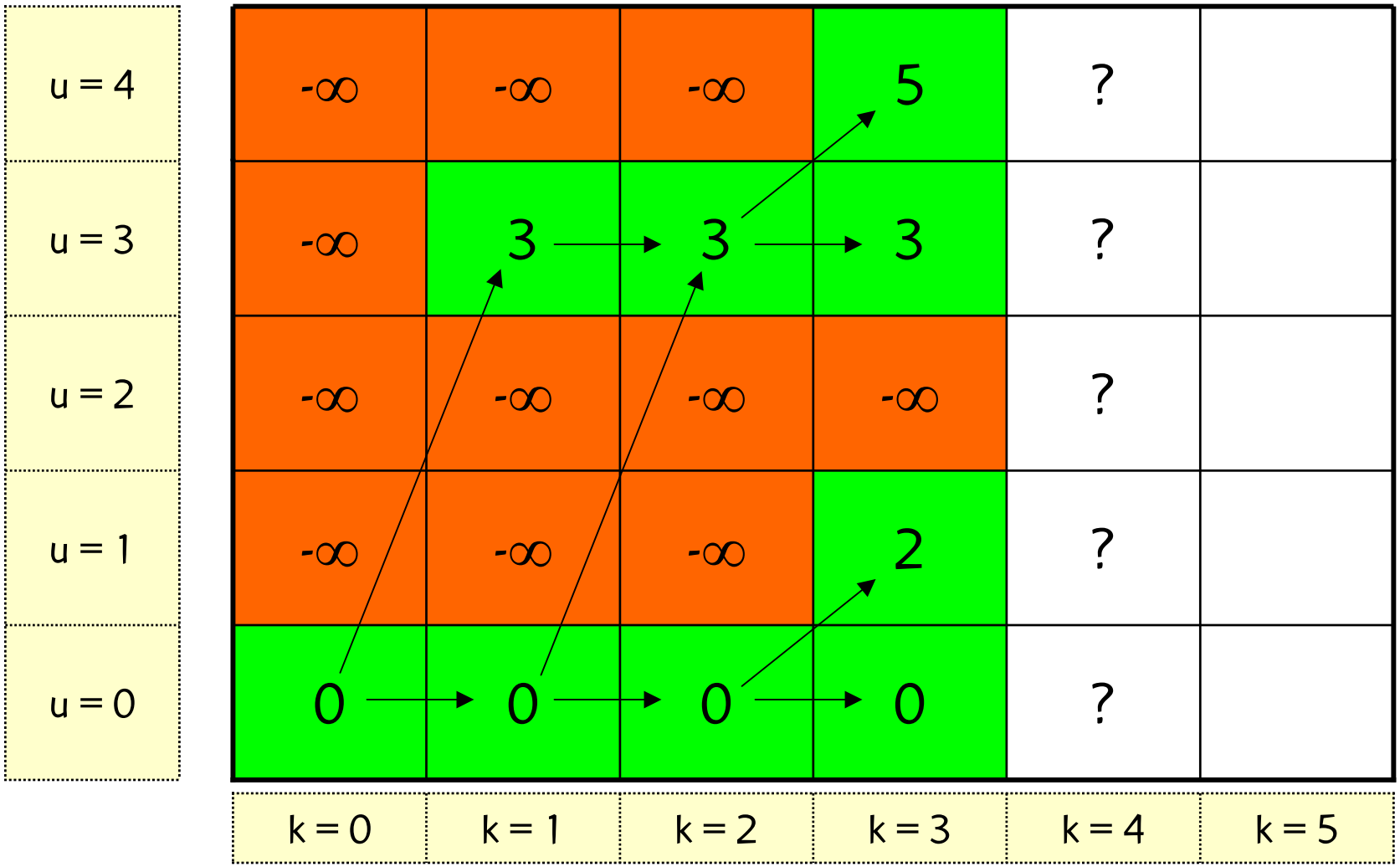
Problema da mochila



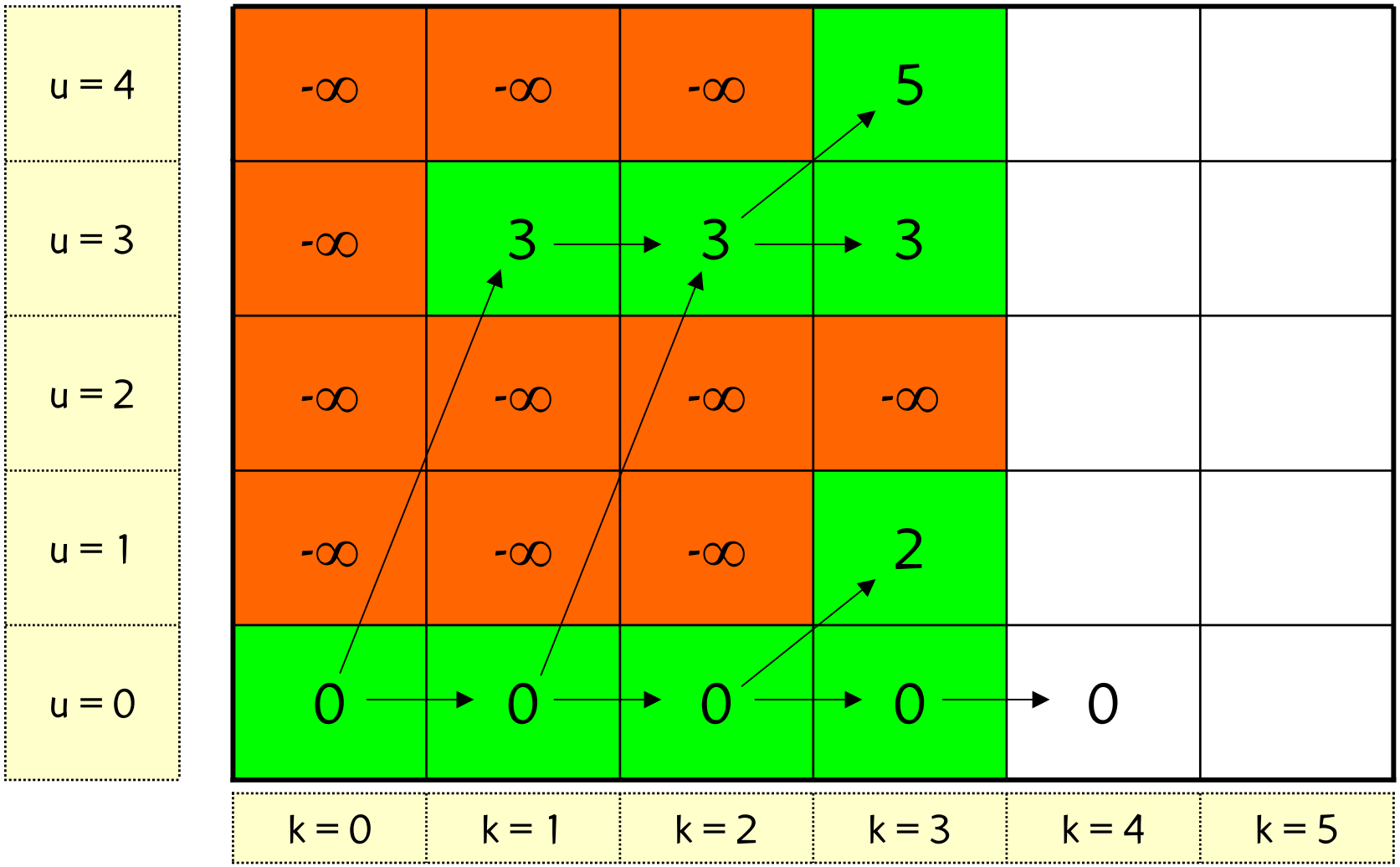
Problema da mochila



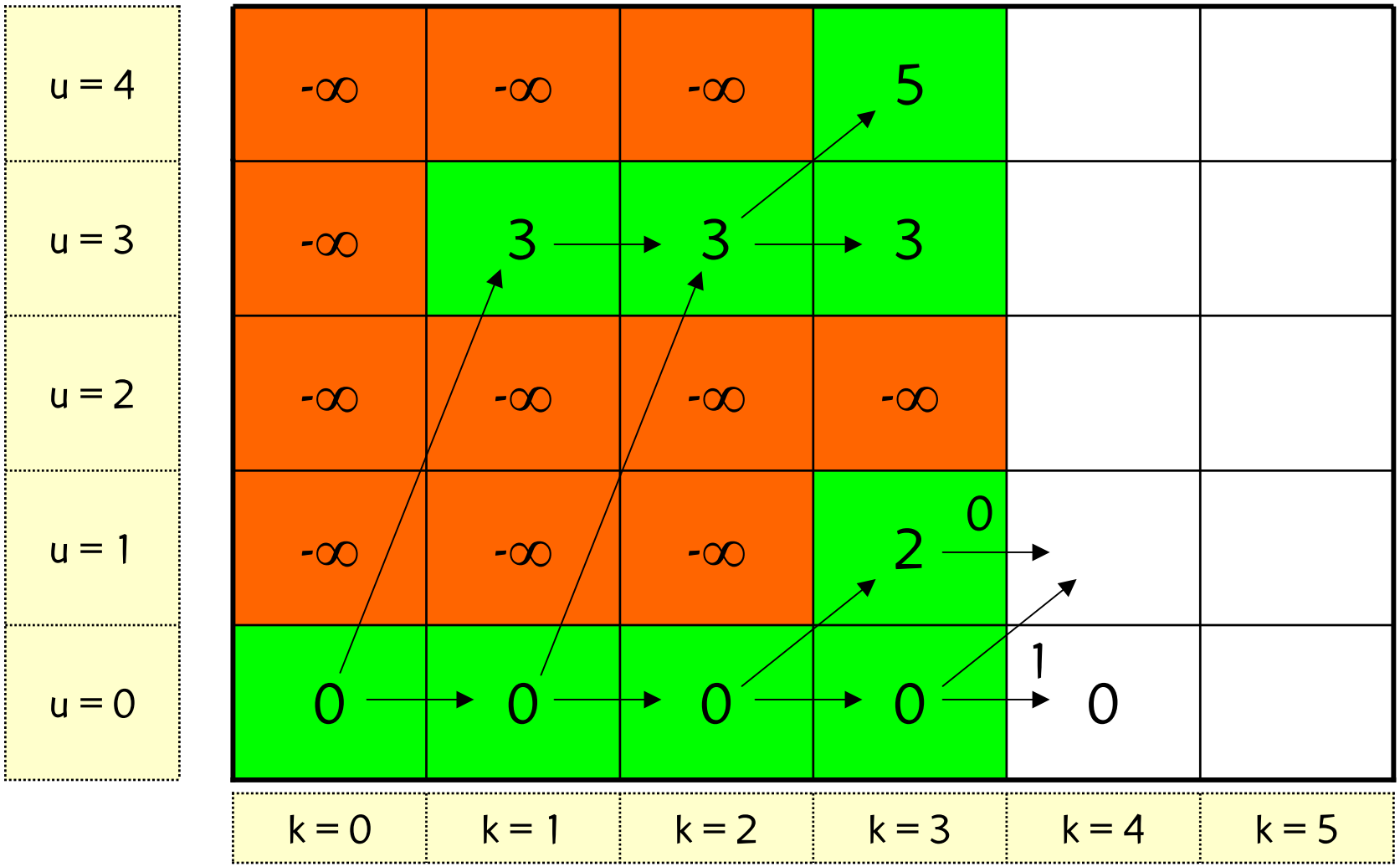
Problema da mochila



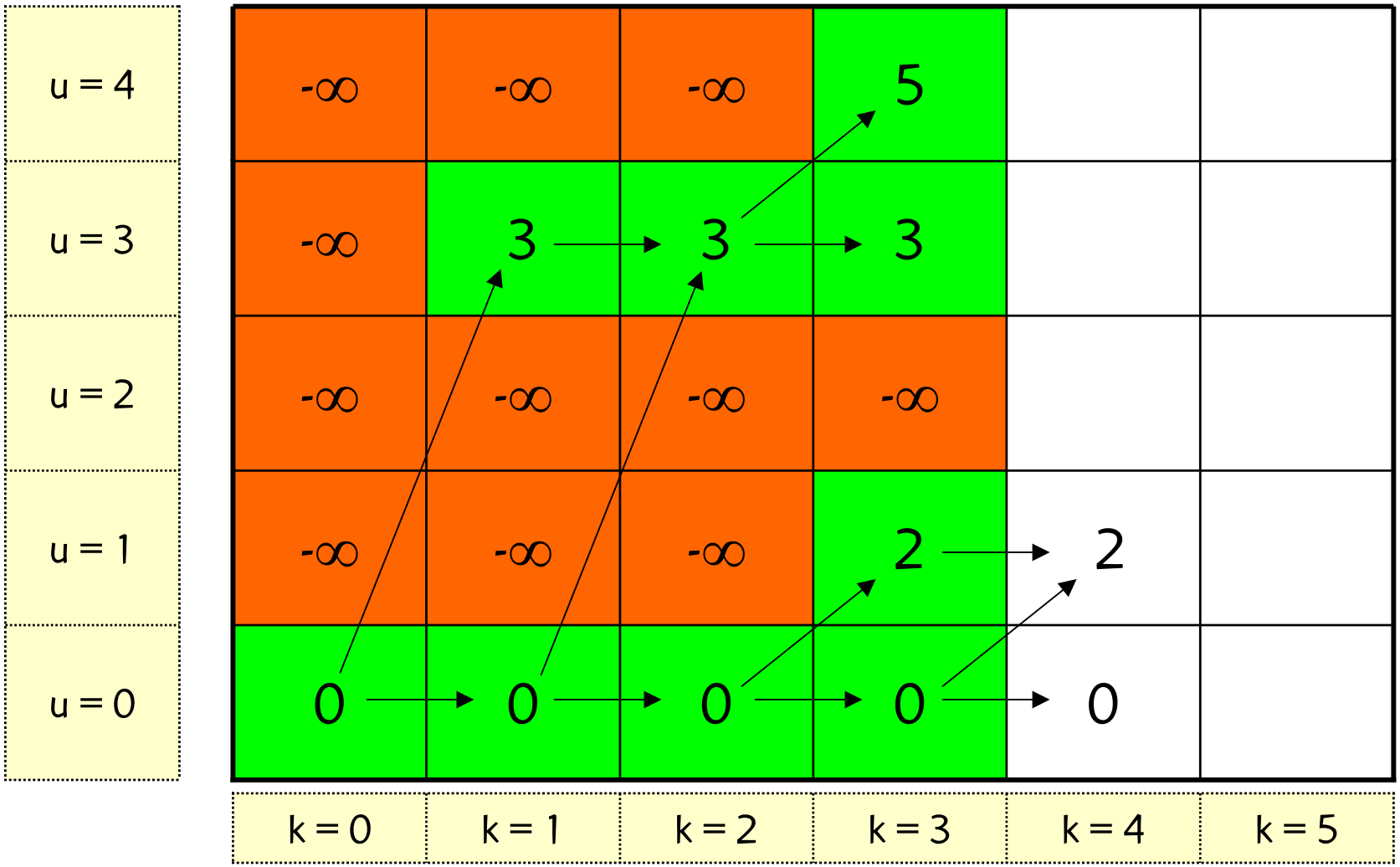
Problema da mochila



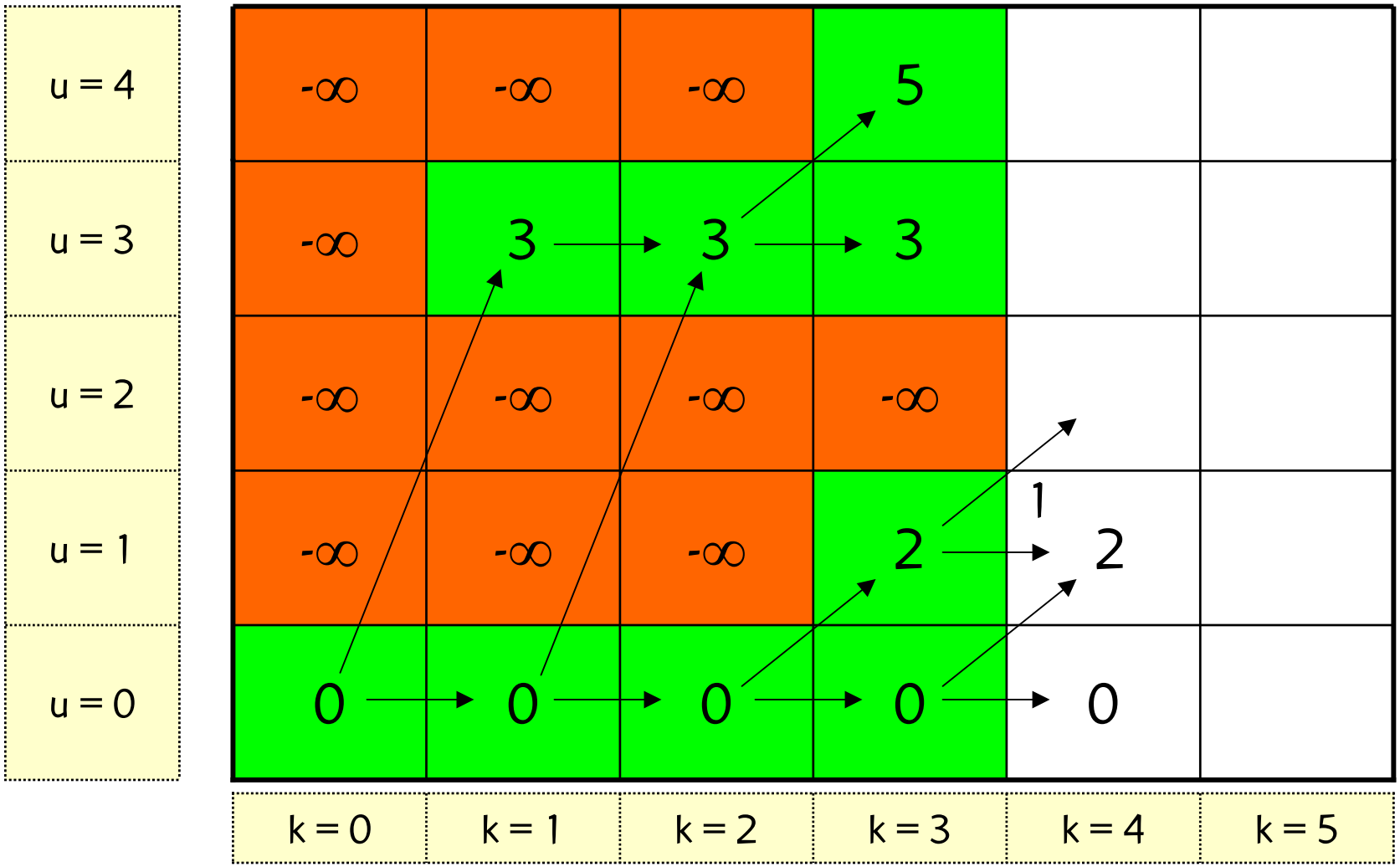
Problema da mochila



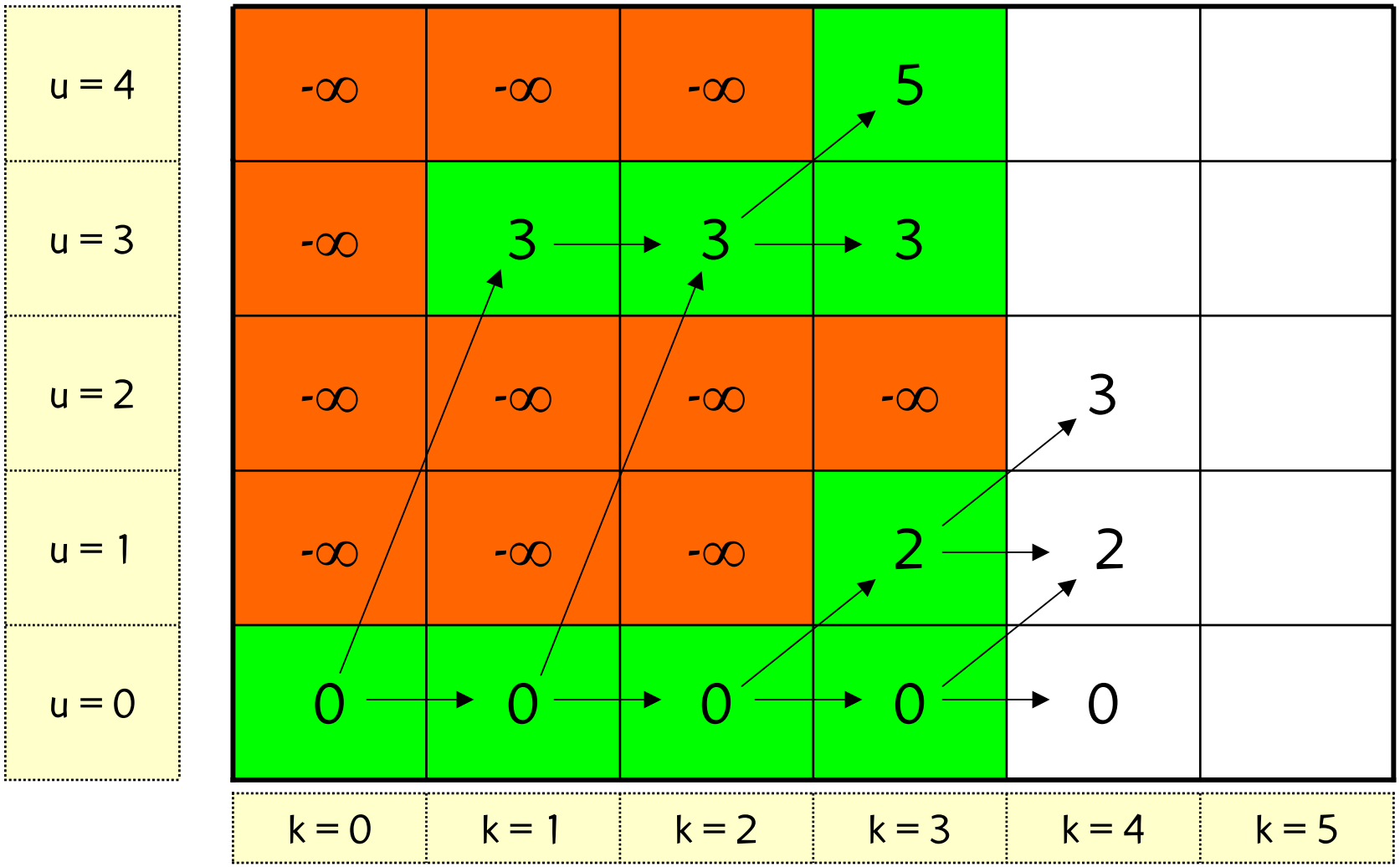
Problema da mochila



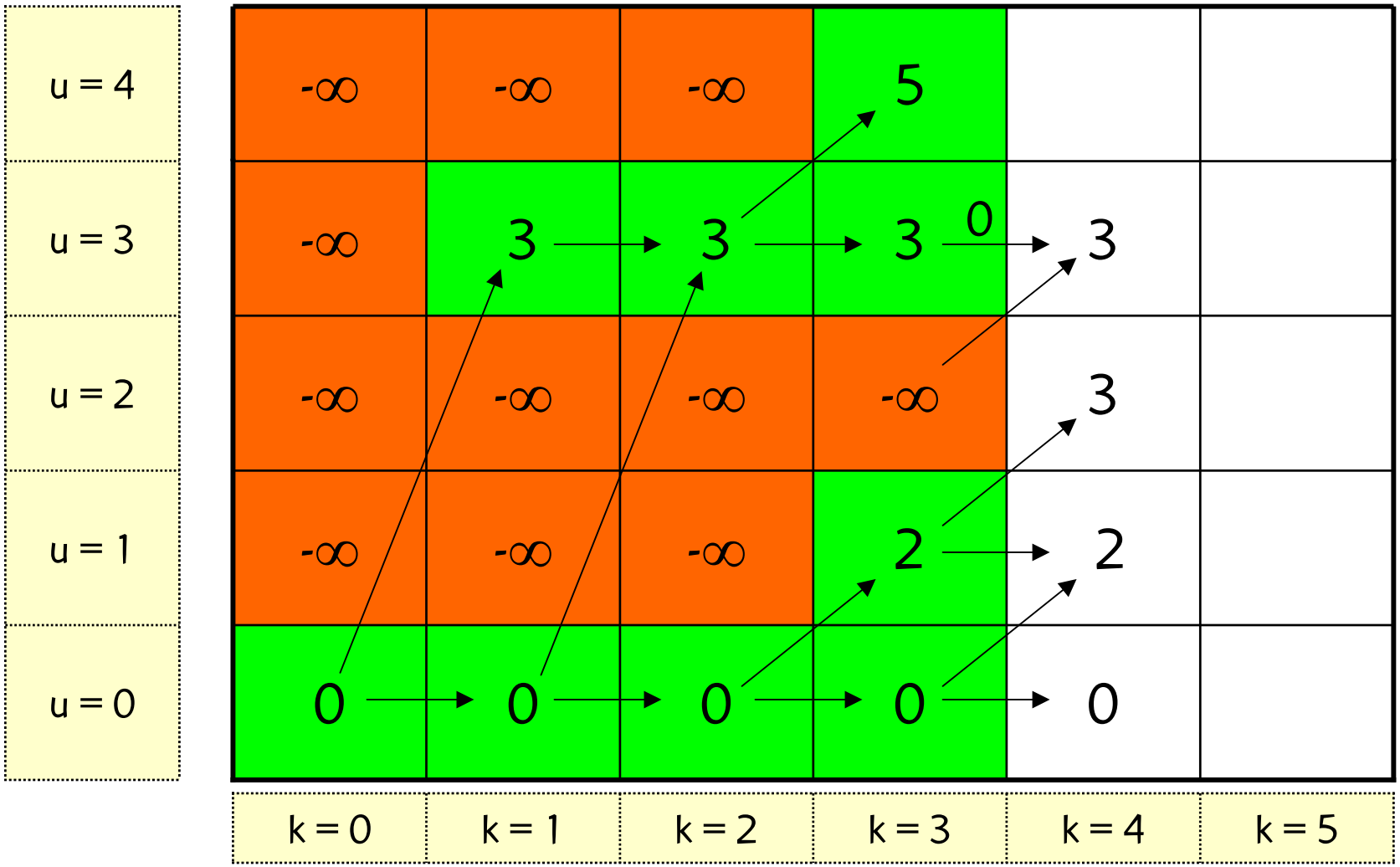
Problema da mochila



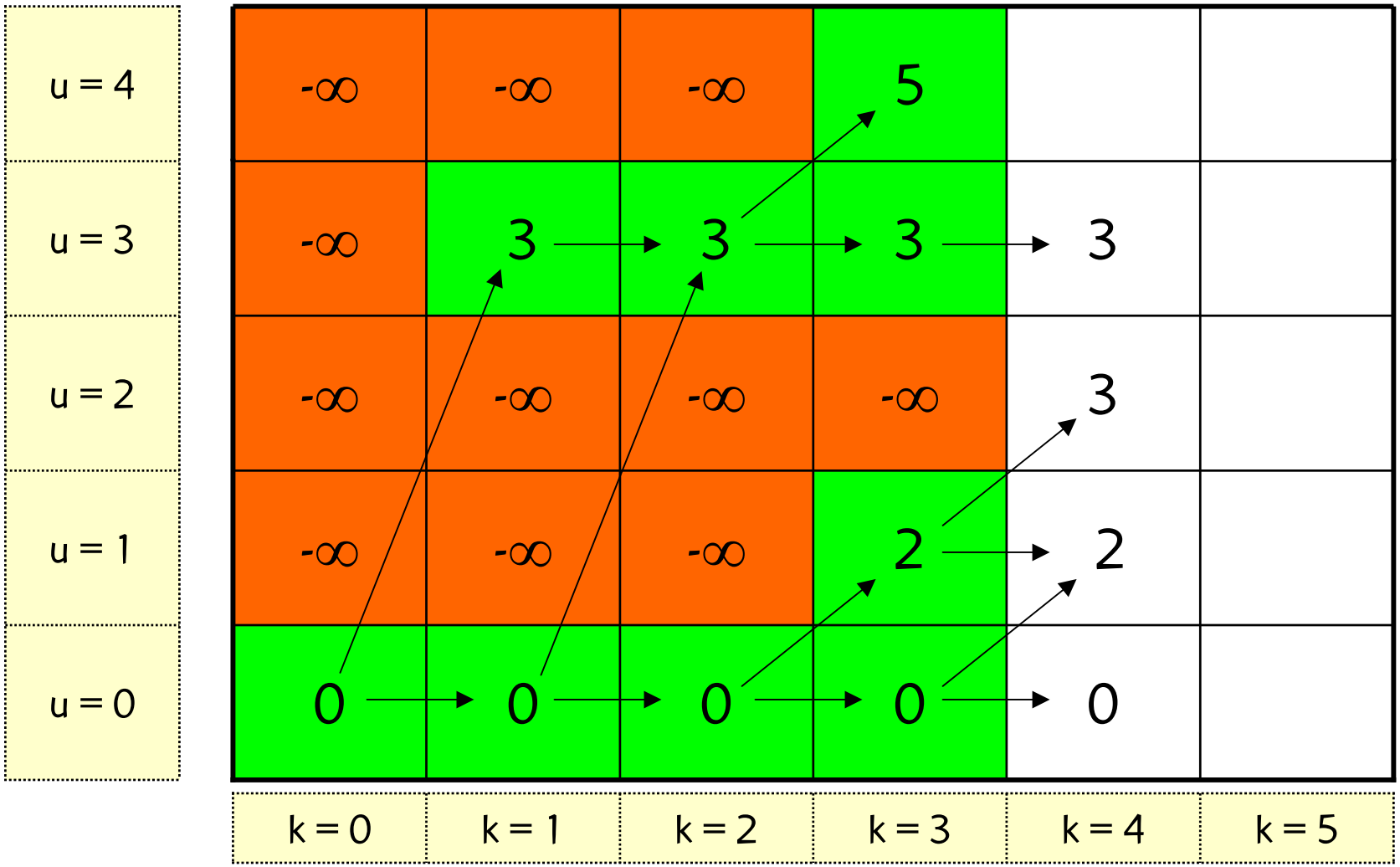
Problema da mochila



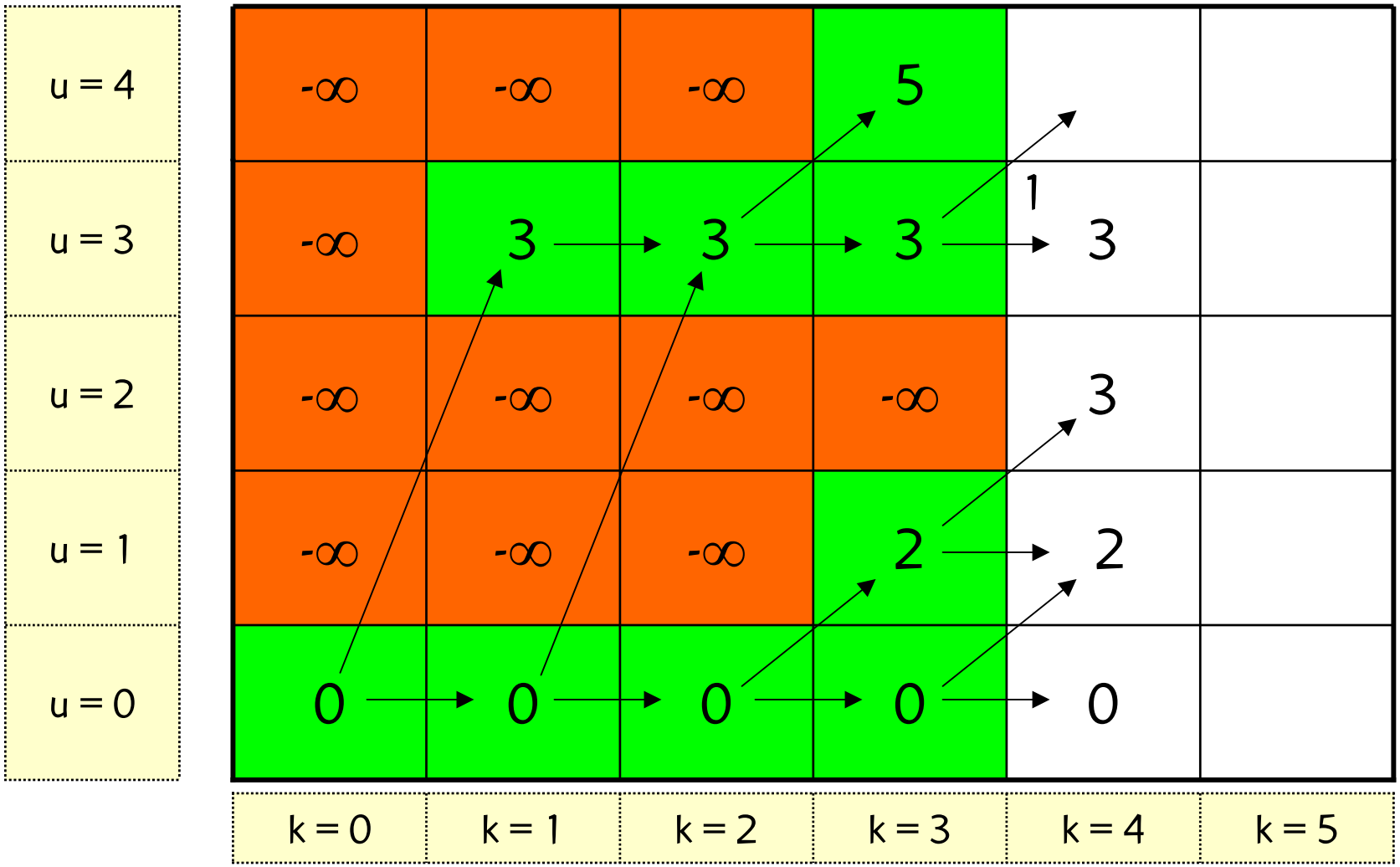
Problema da mochila



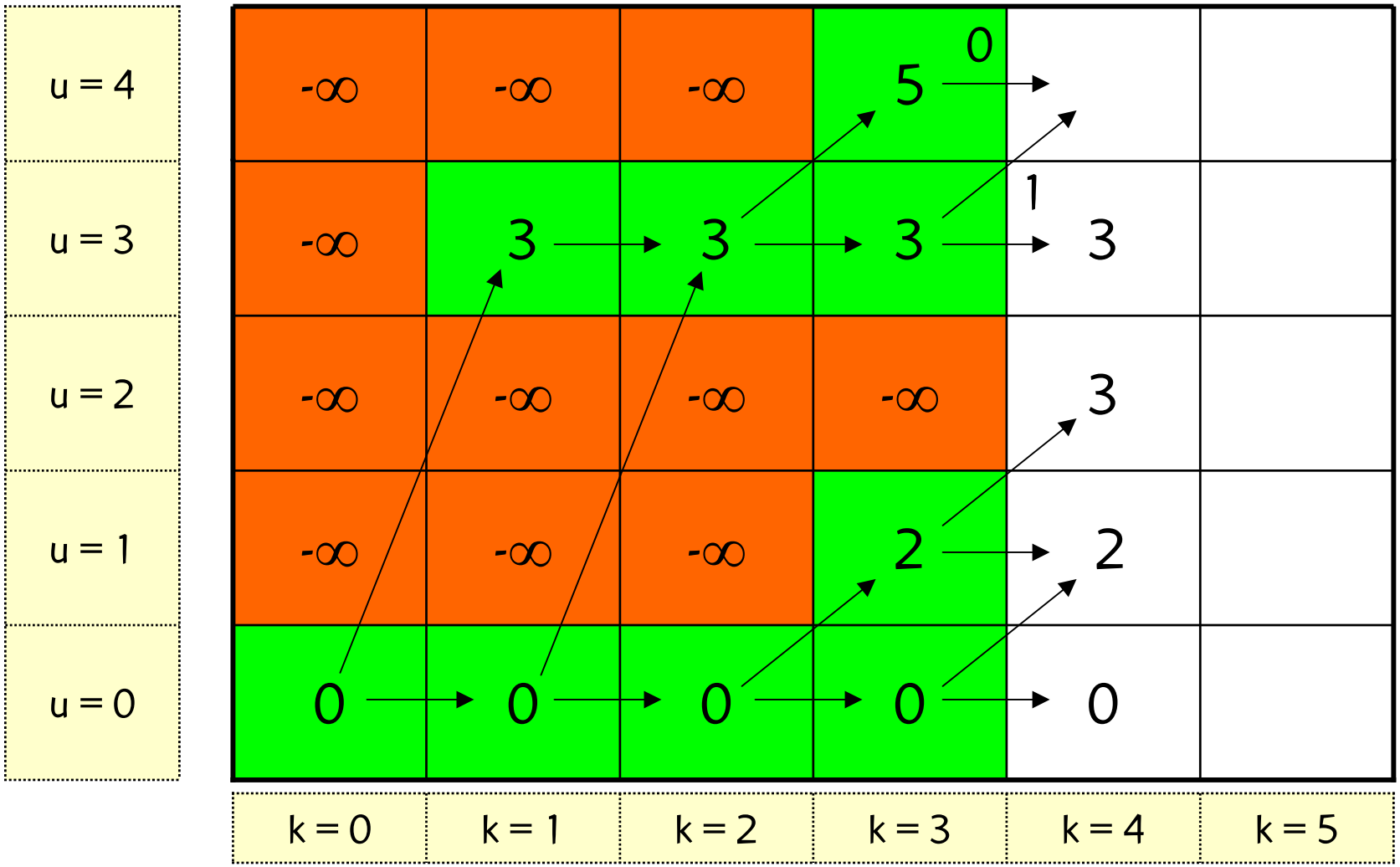
Problema da mochila



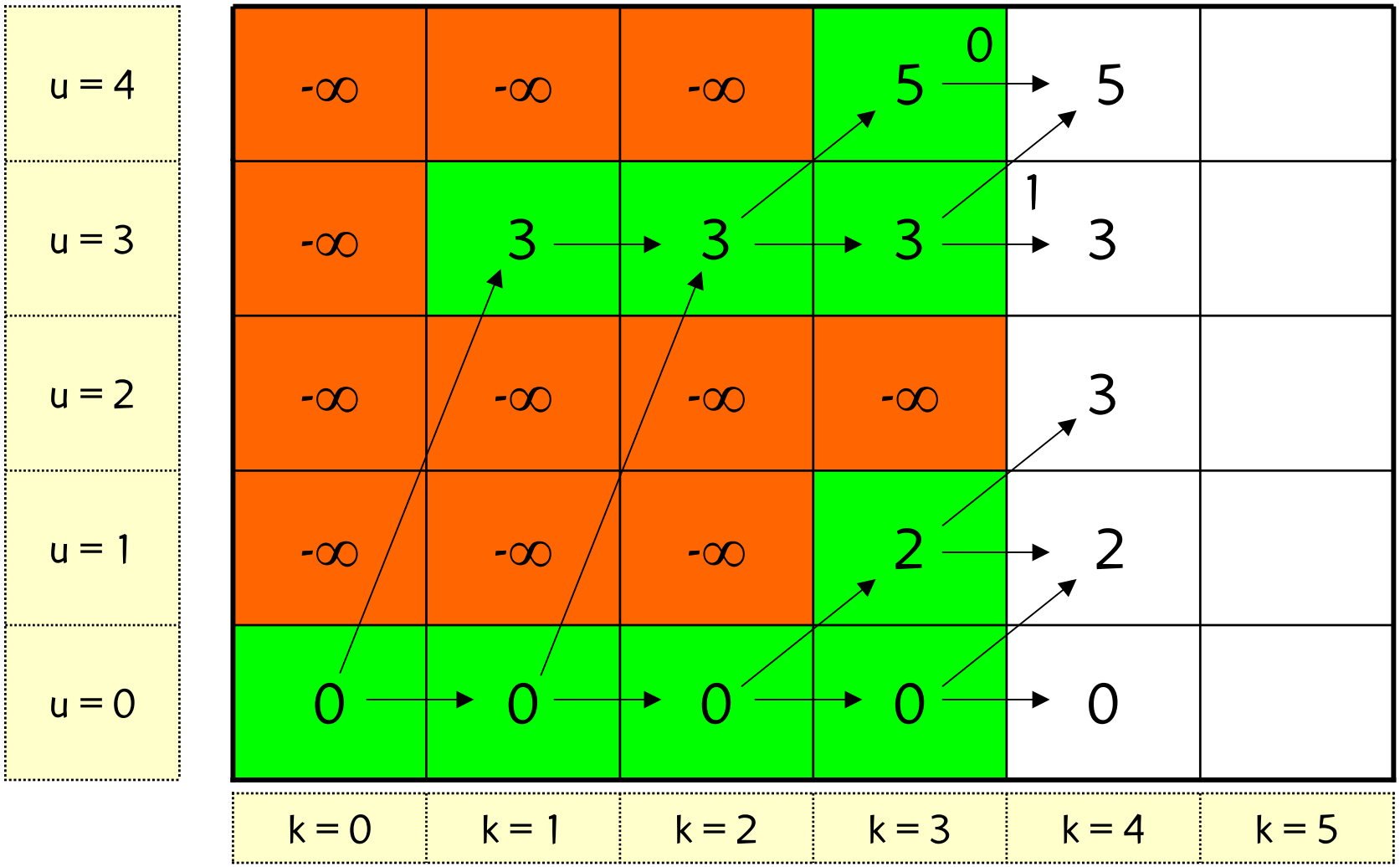
Problema da mochila



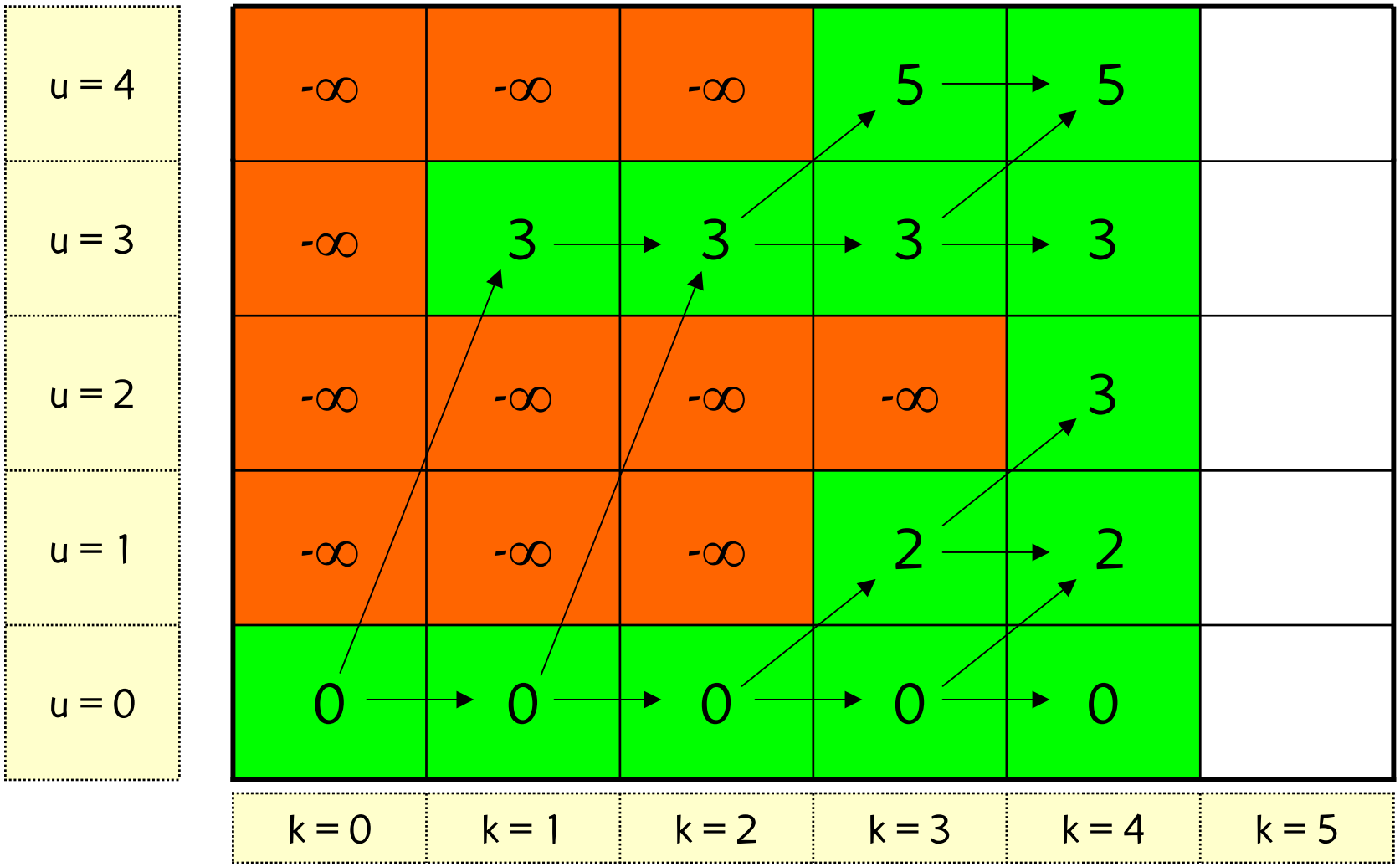
Problema da mochila



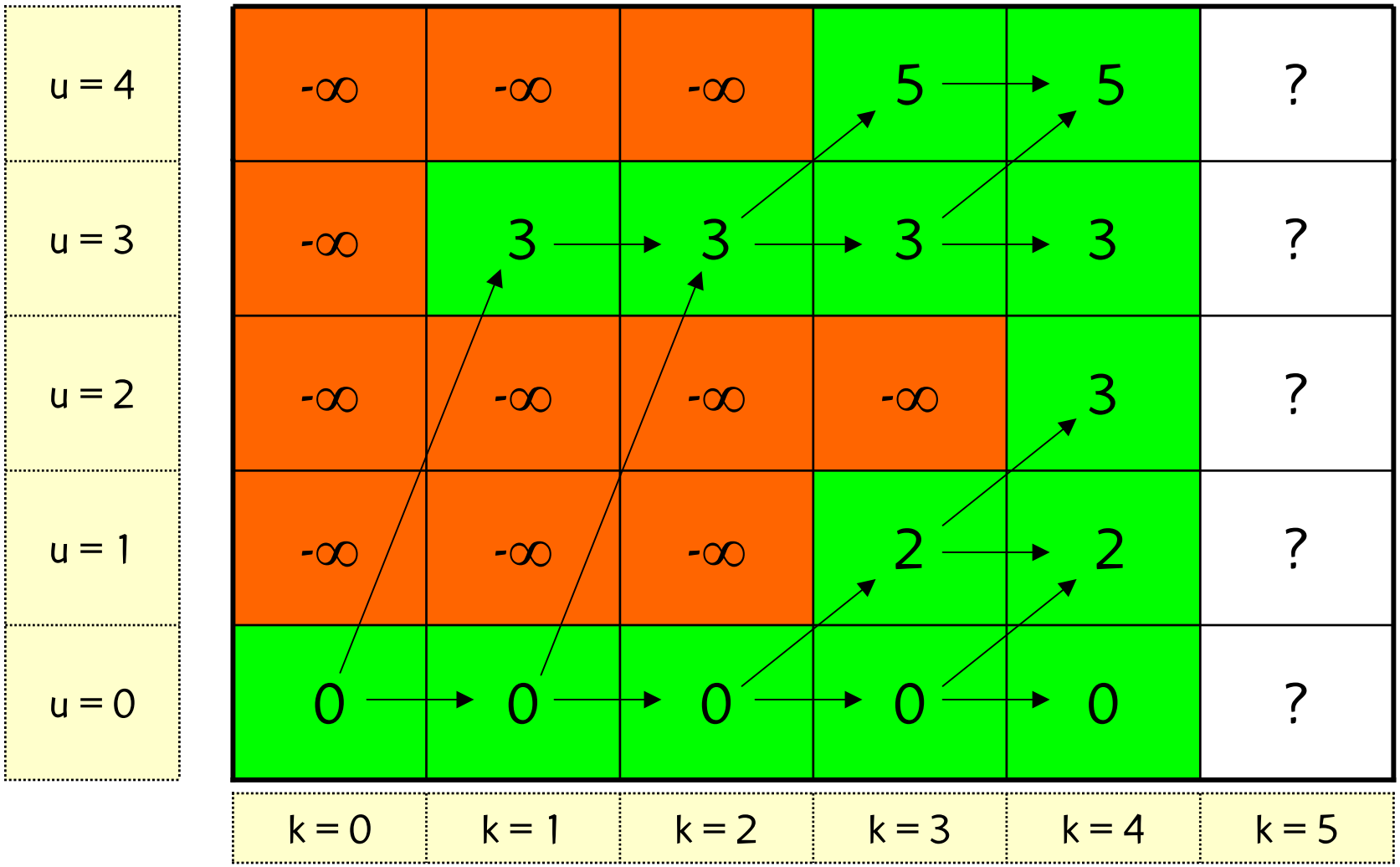
Problema da mochila



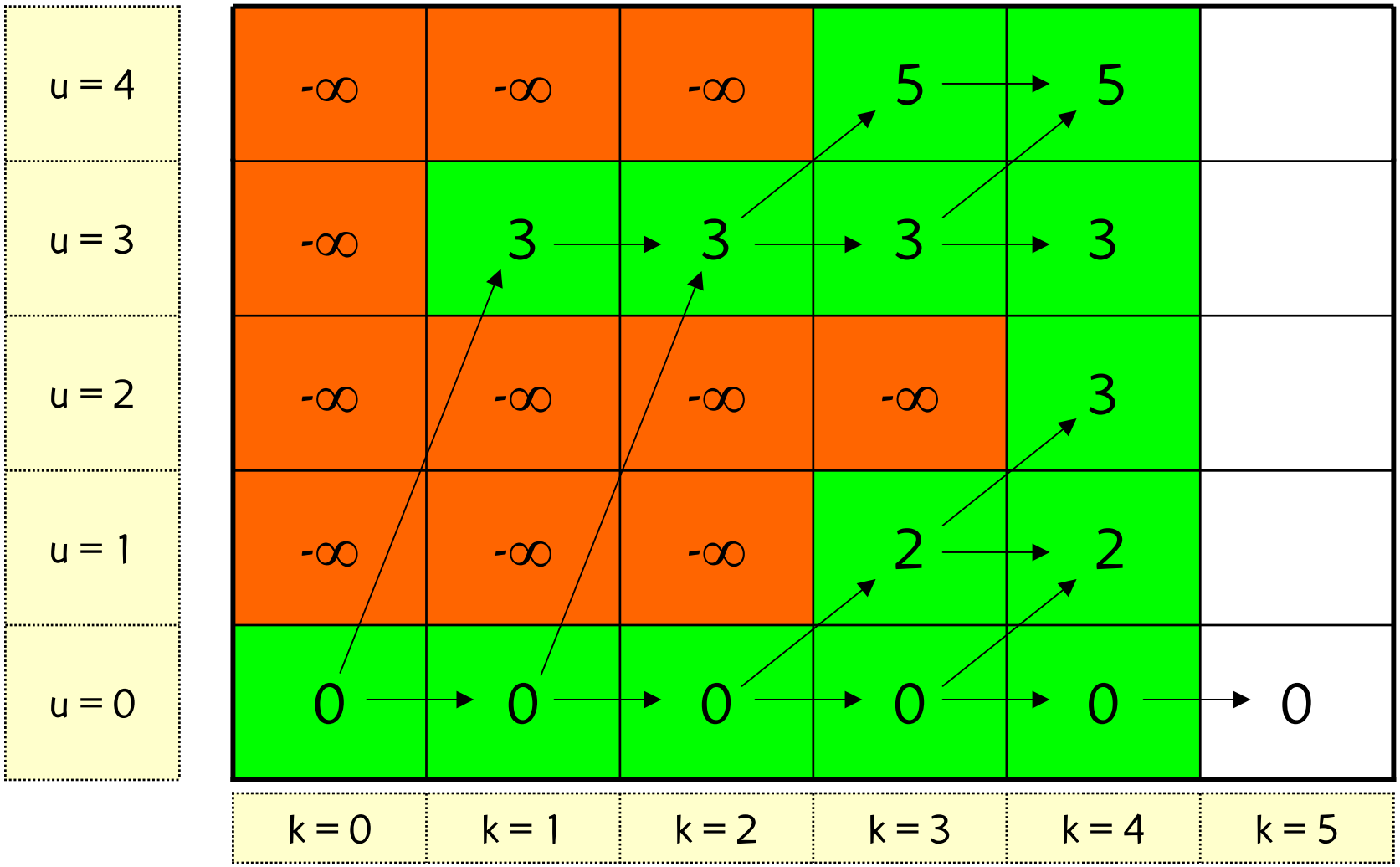
Problema da mochila



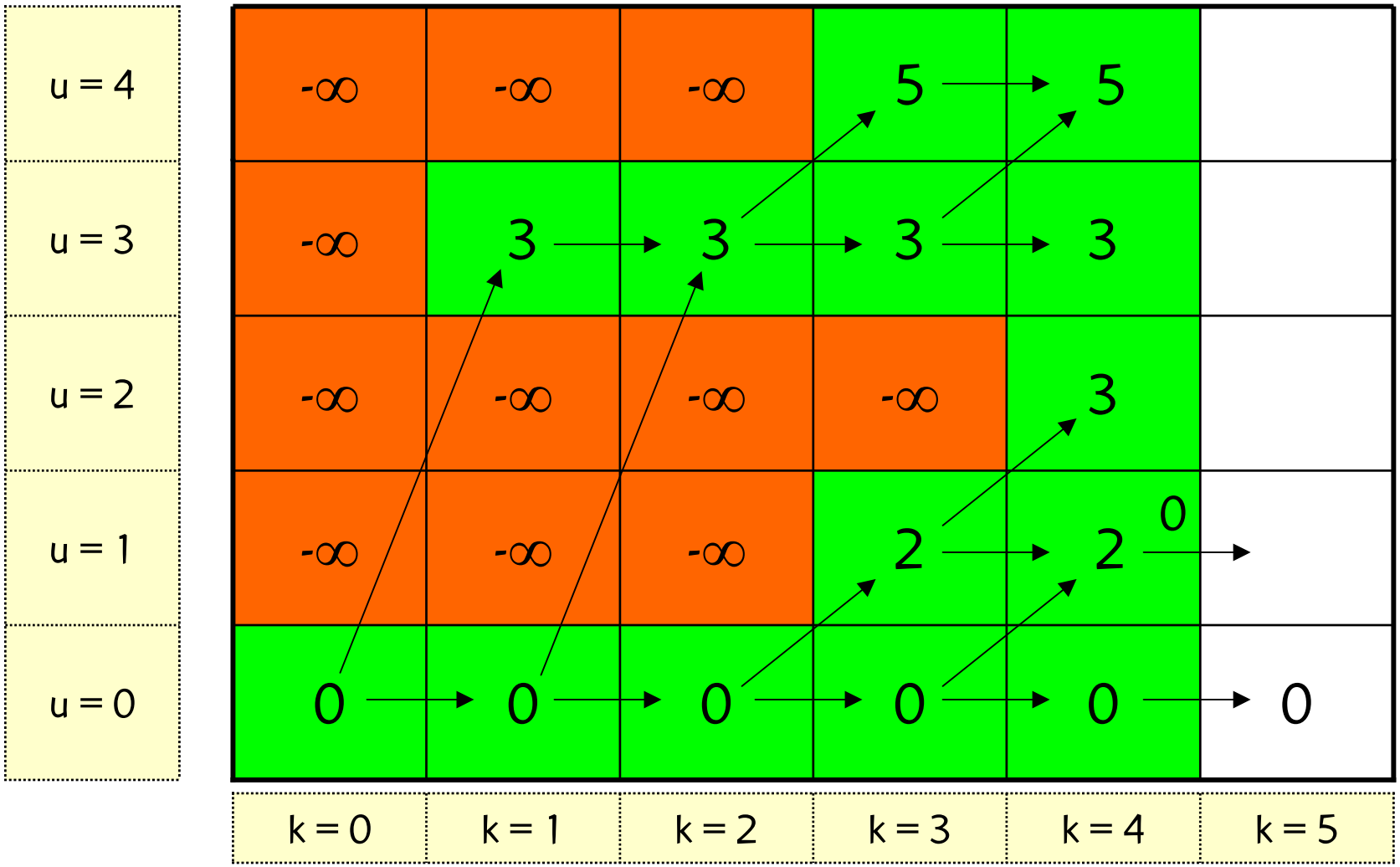
Problema da mochila



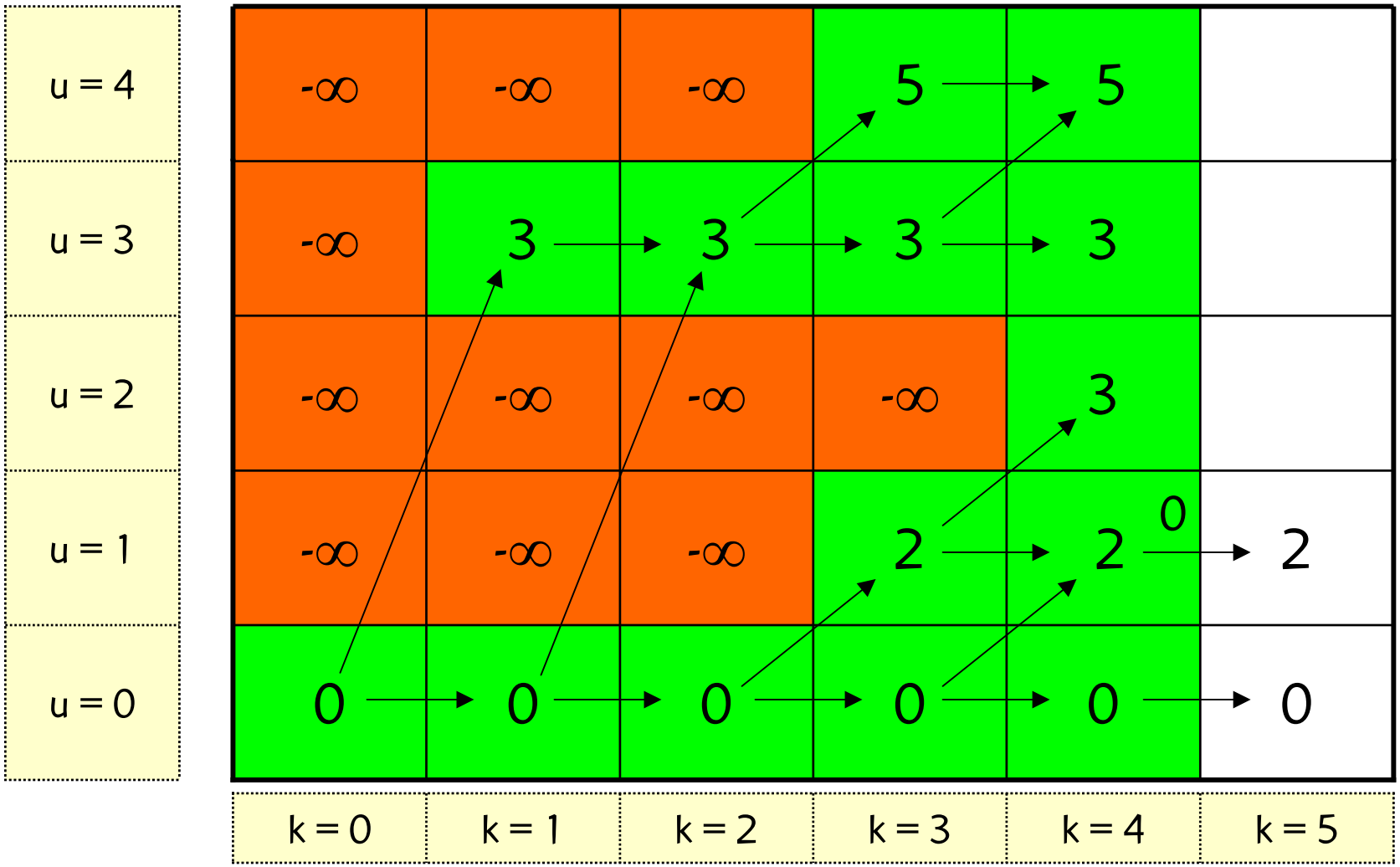
Problema da mochila



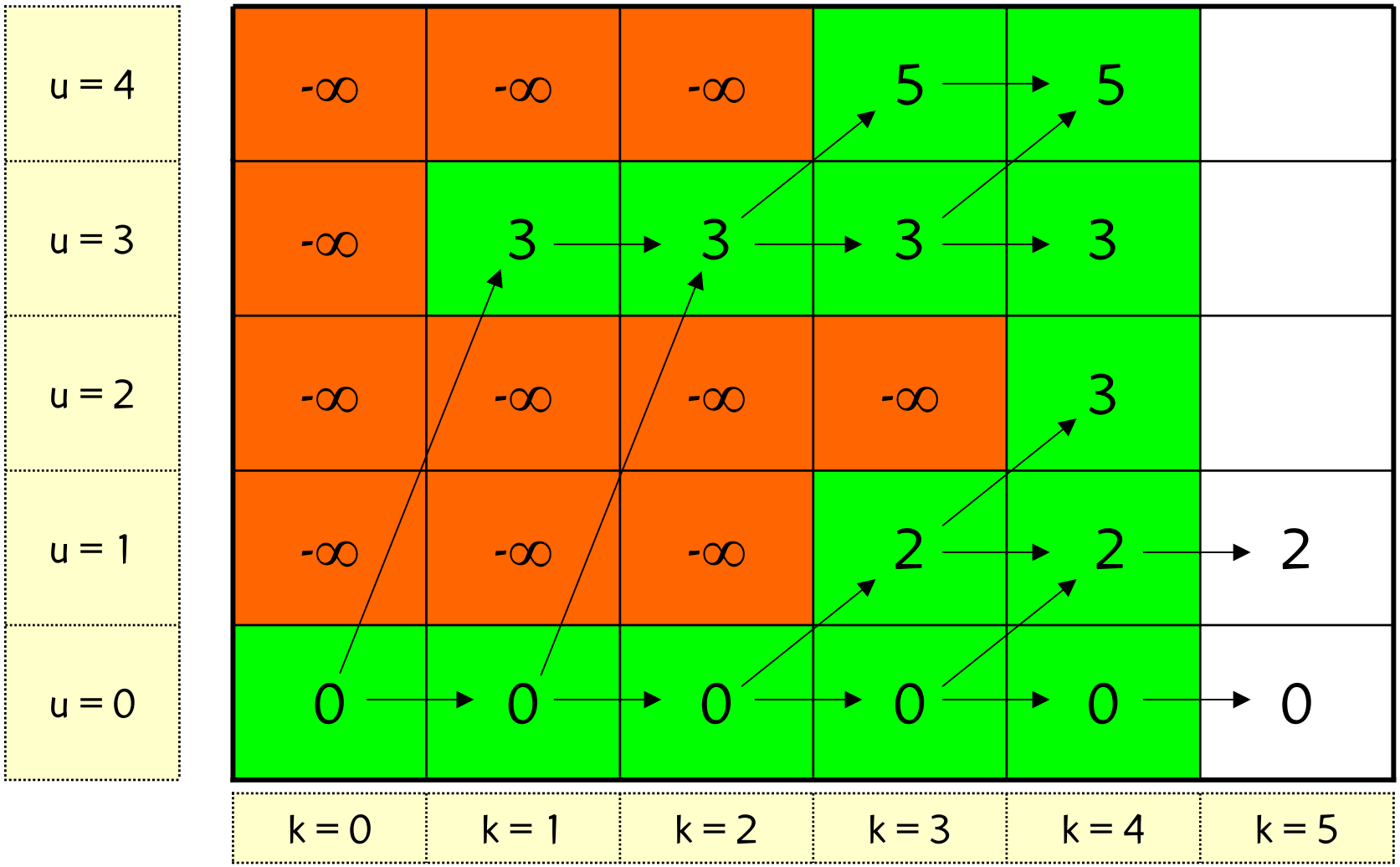
Problema da mochila



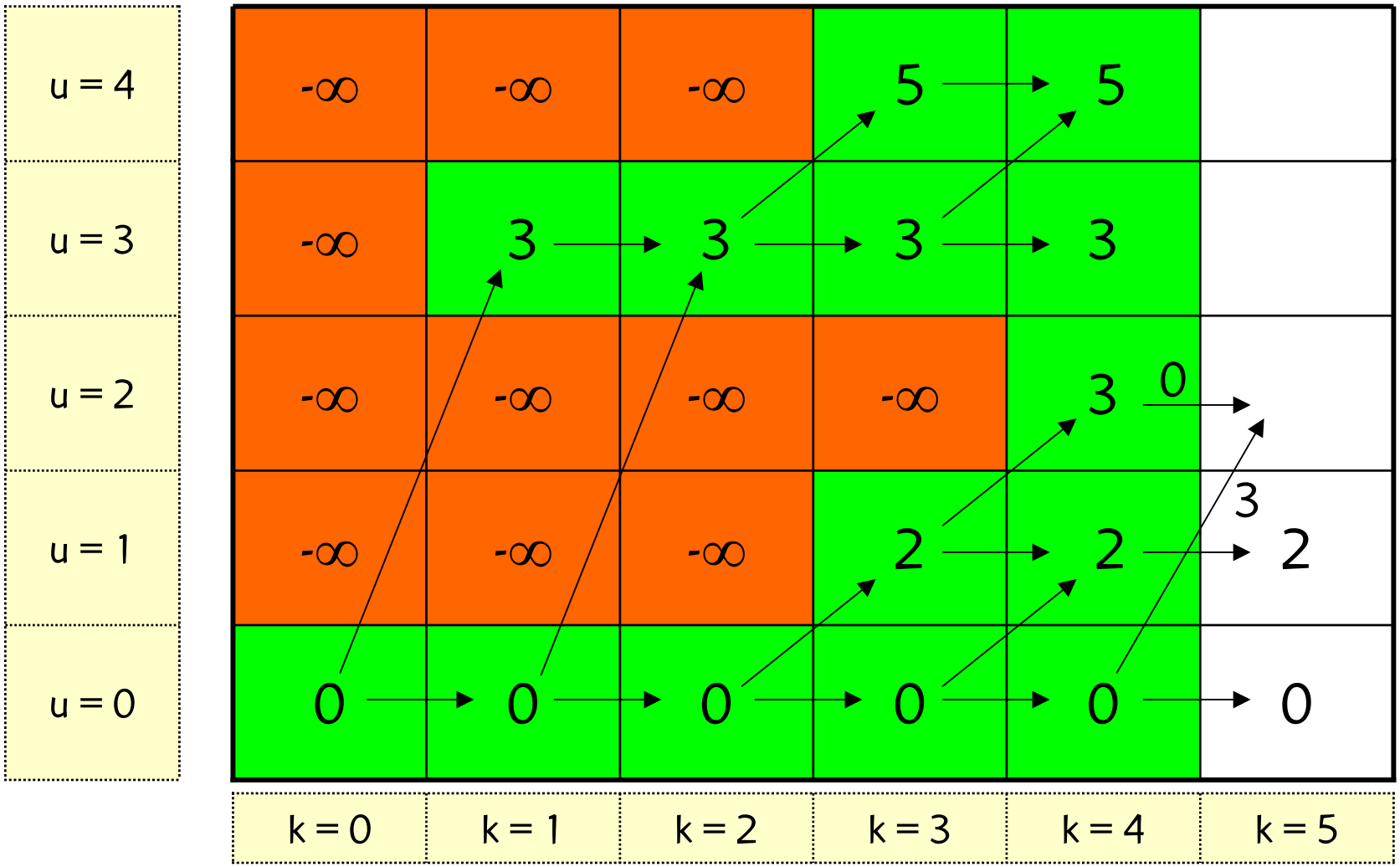
Problema da mochila



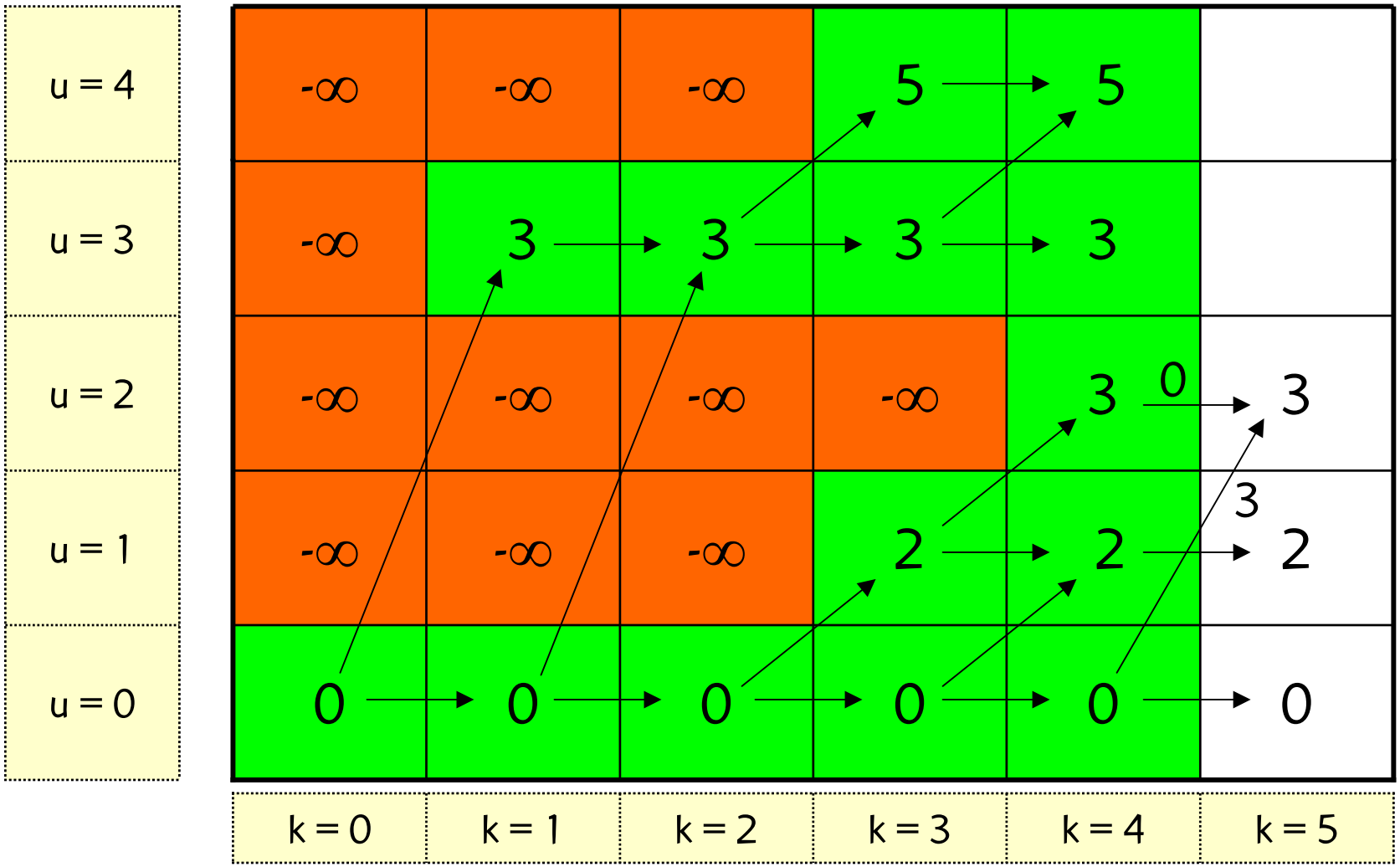
Problema da mochila



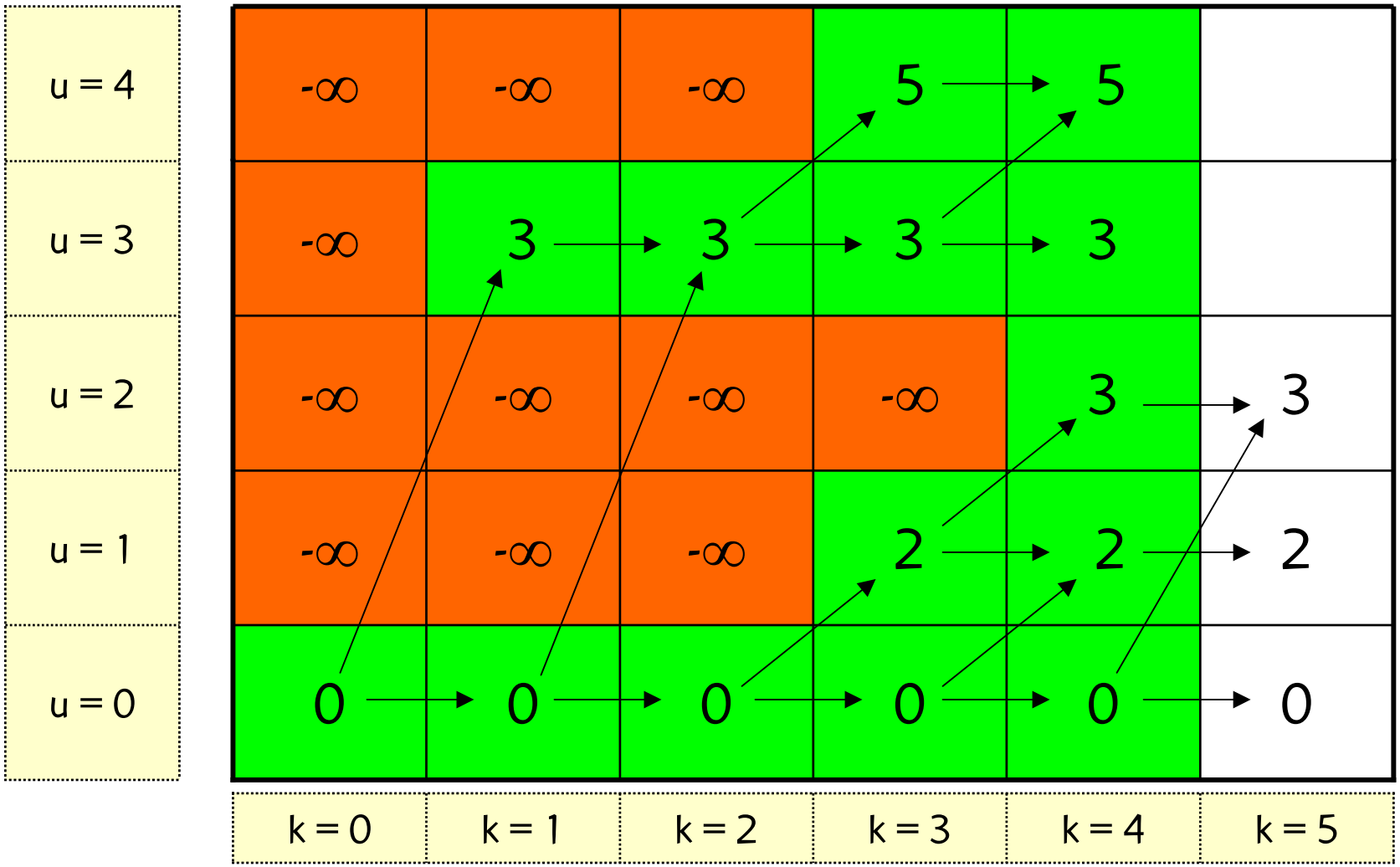
Problema da mochila



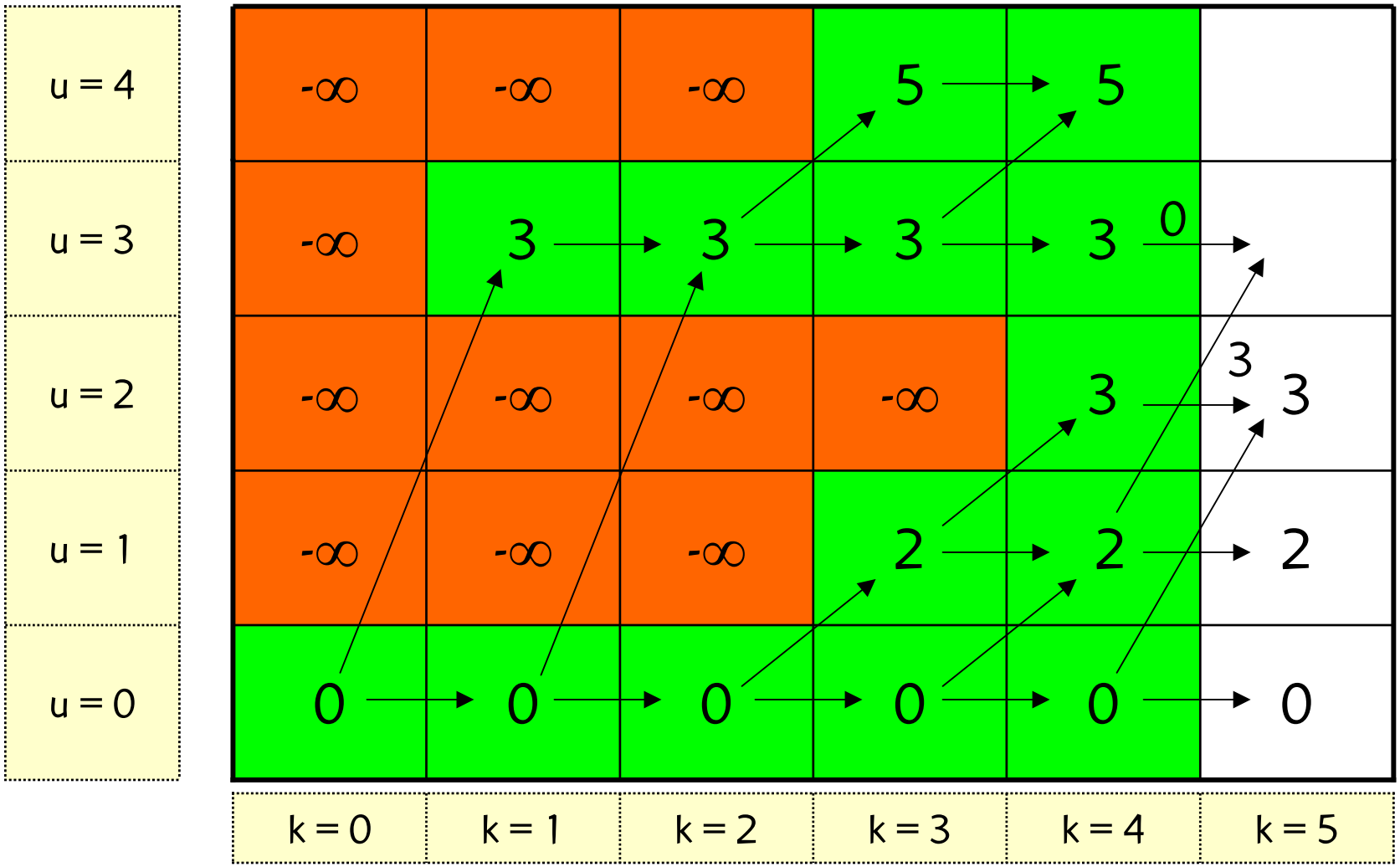
Problema da mochila



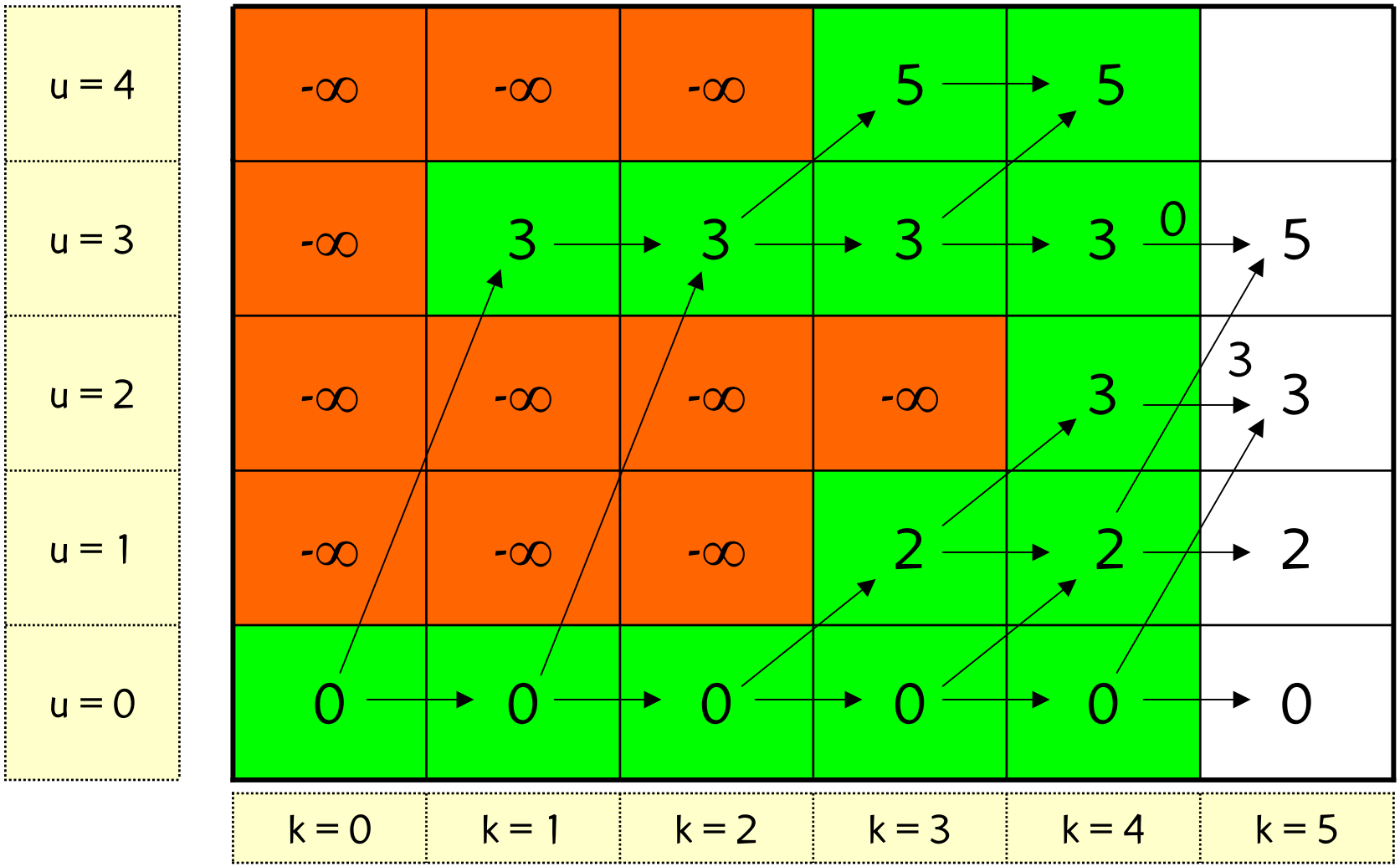
Problema da mochila



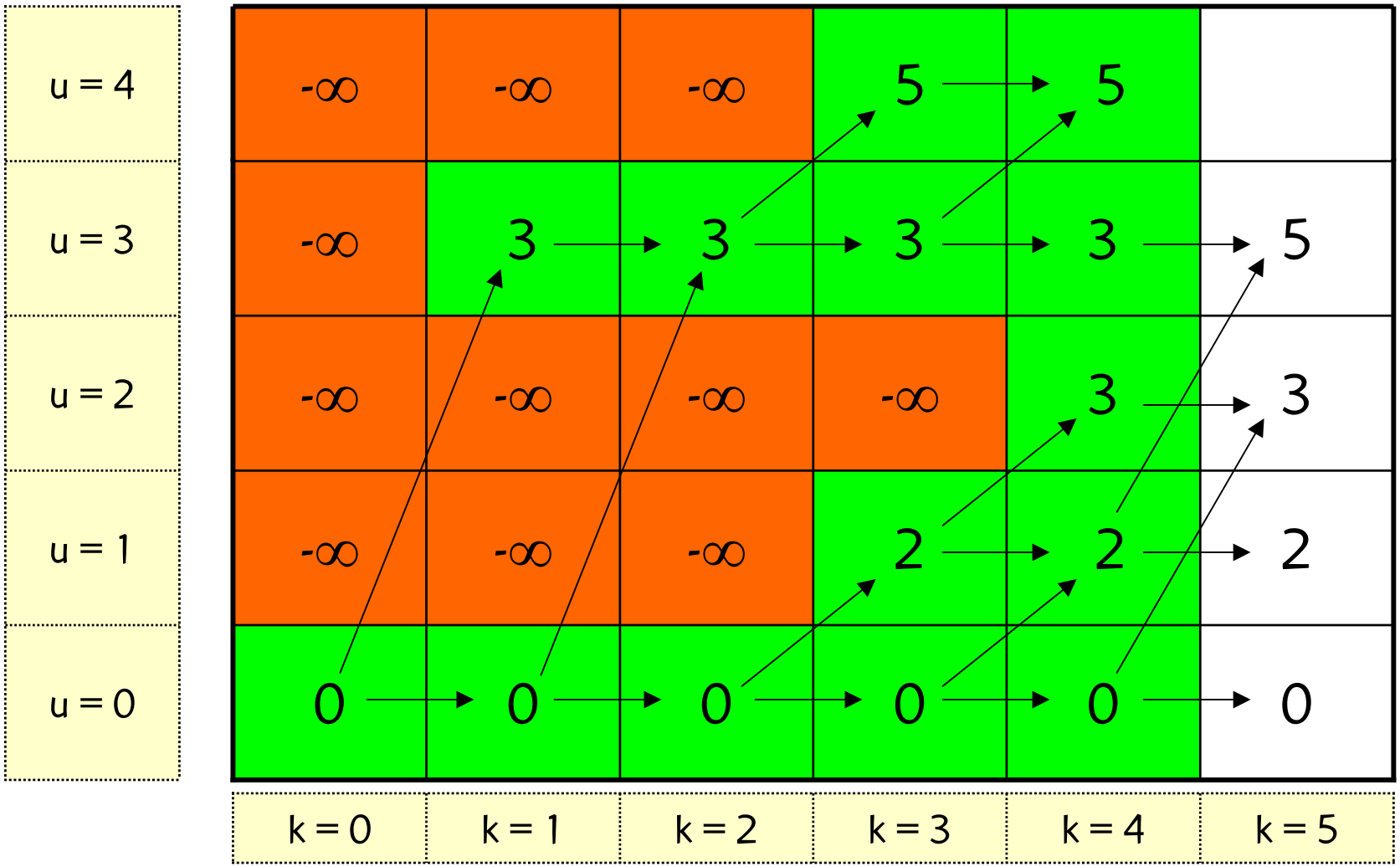
Problema da mochila



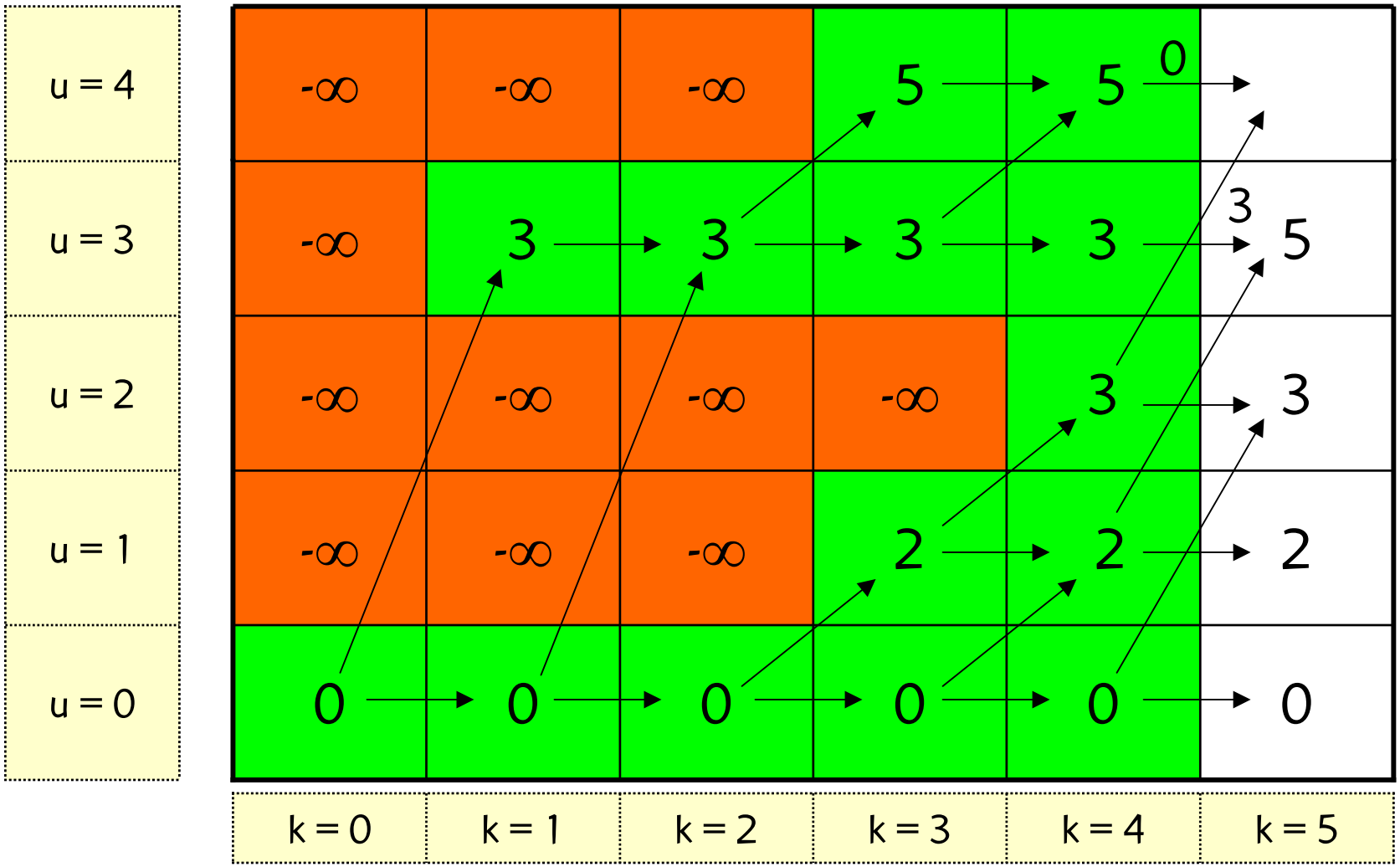
Problema da mochila



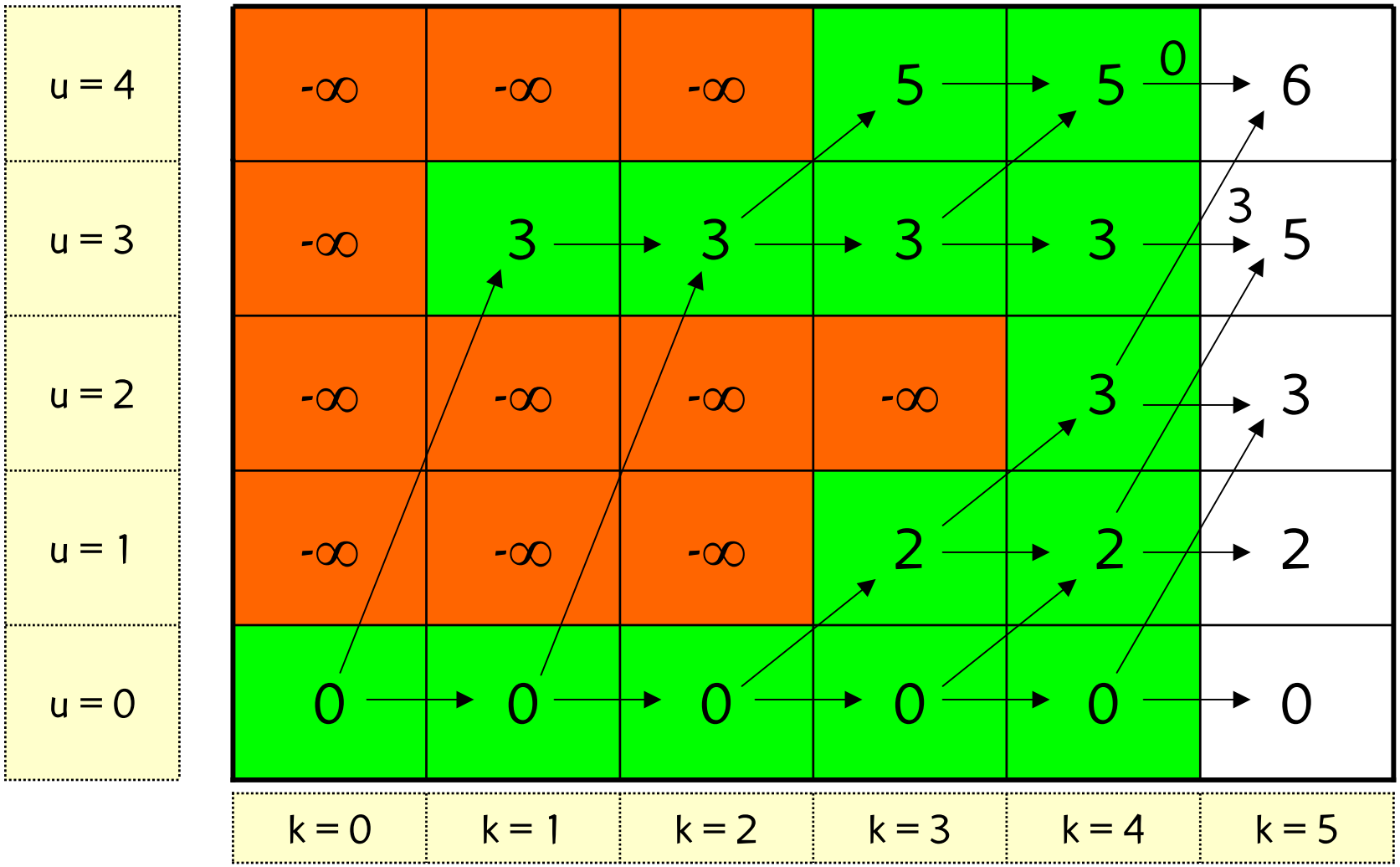
Problema da mochila



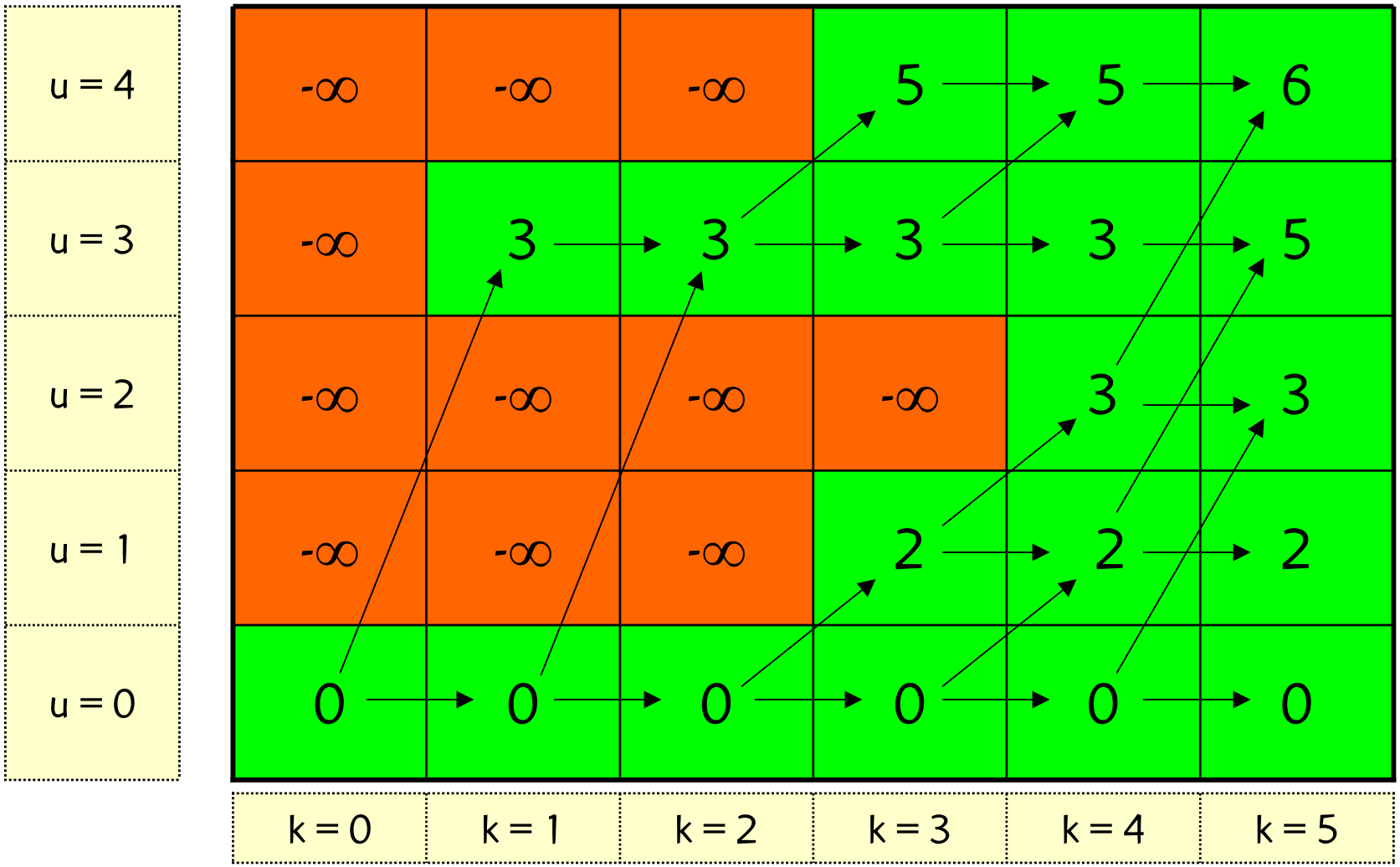
Problema da mochila



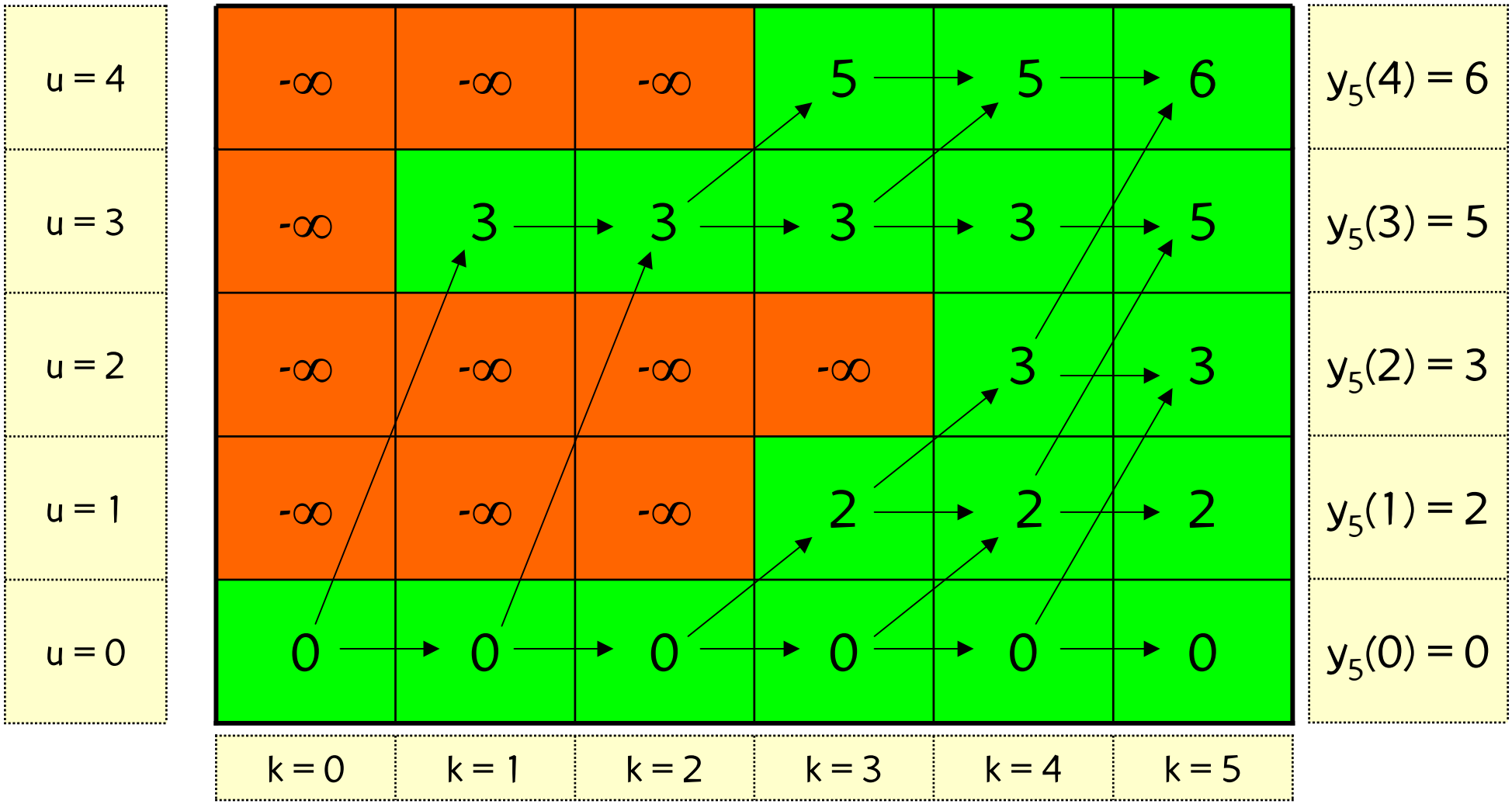
Problema da mochila



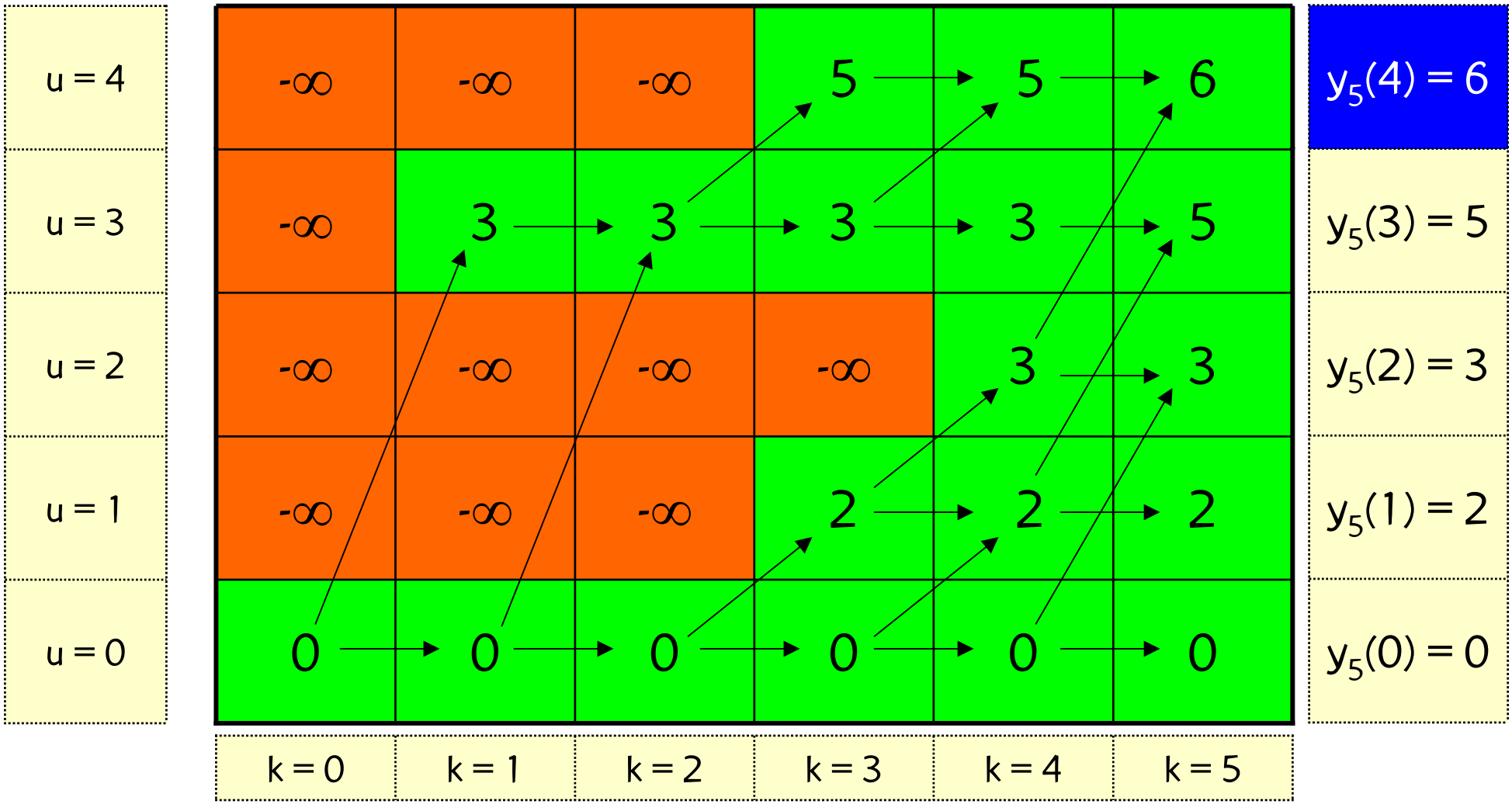
Problema da mochila



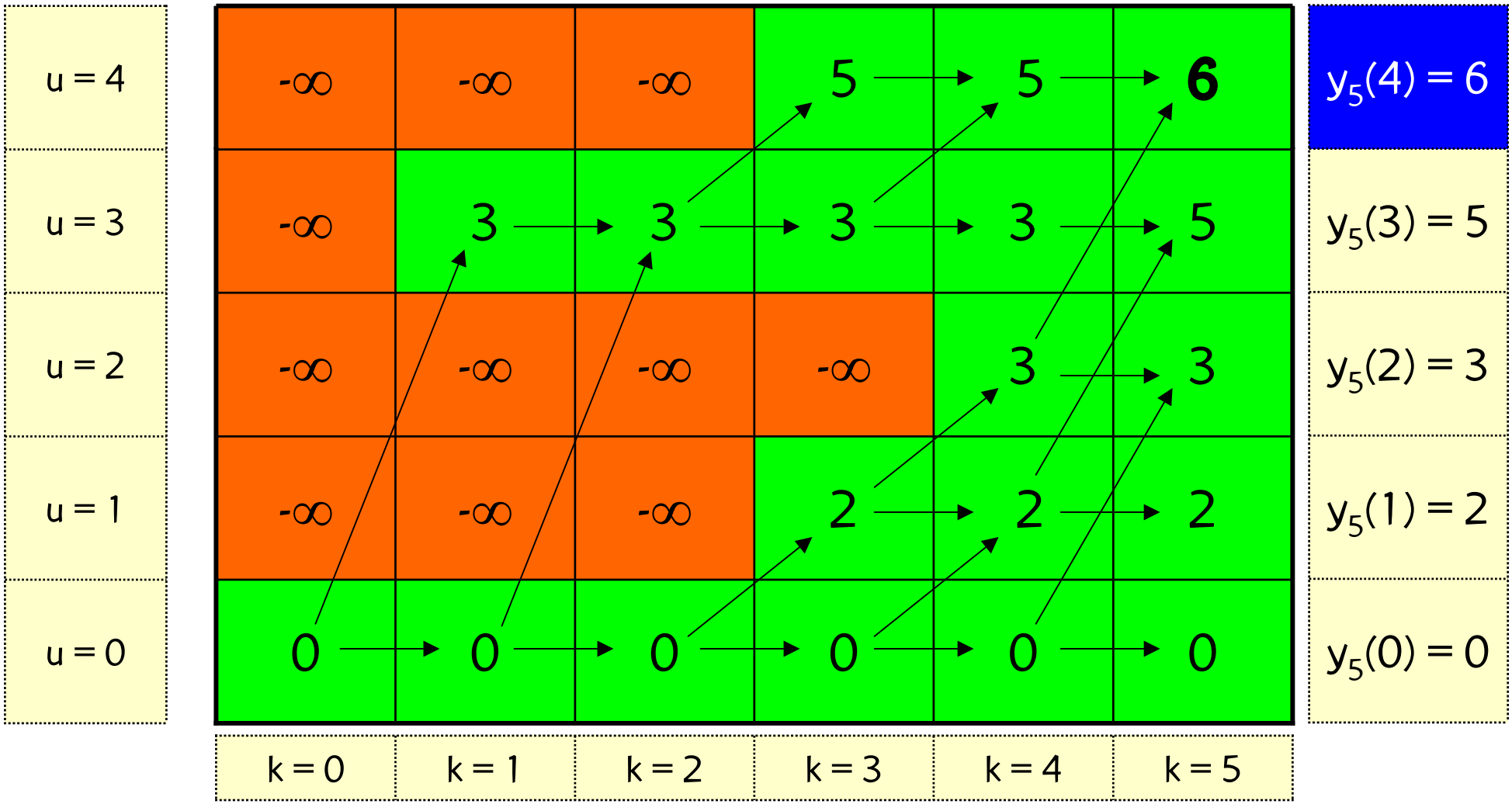
Problema da mochila



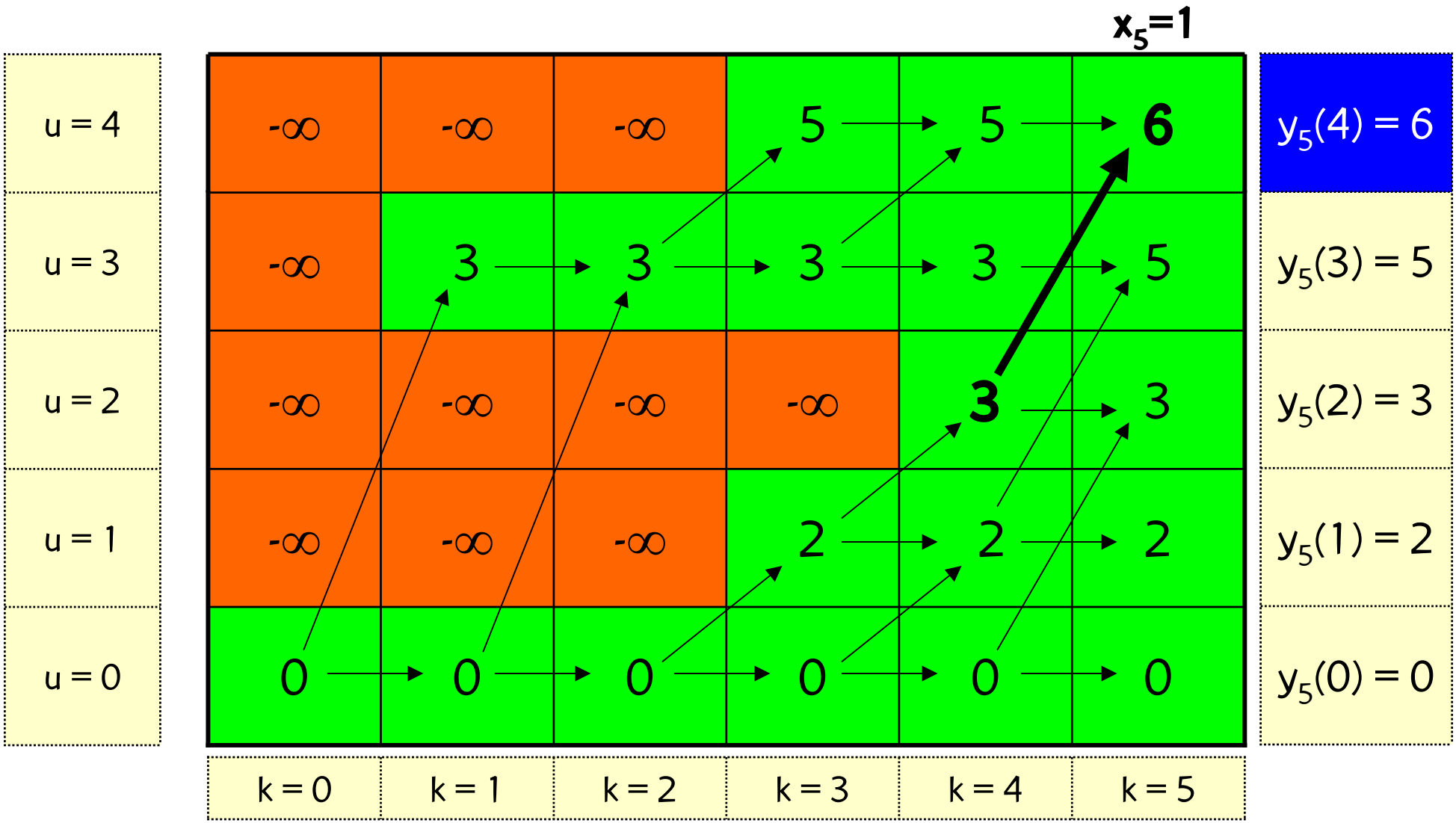
Problema da mochila



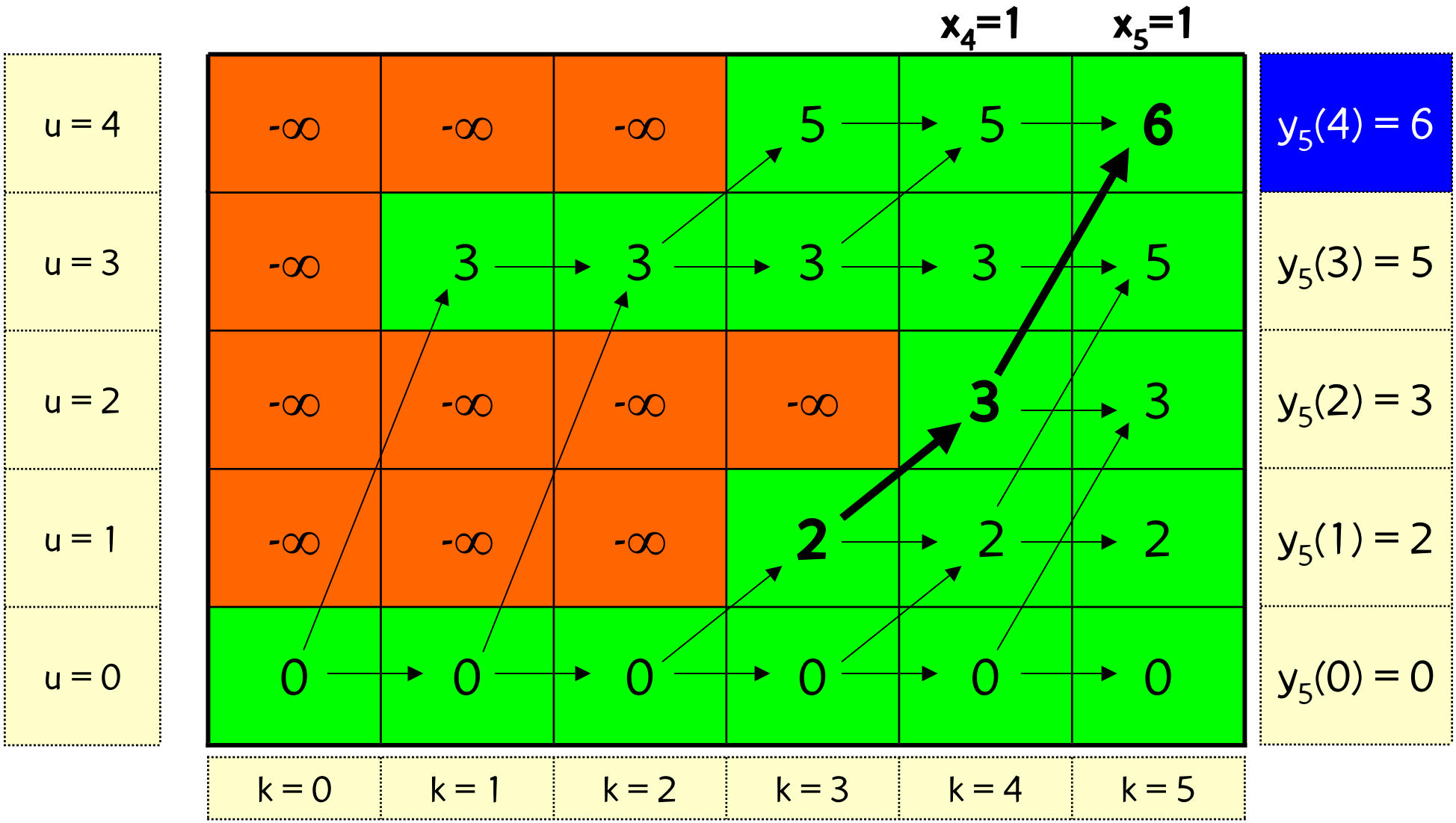
Problema da mochila



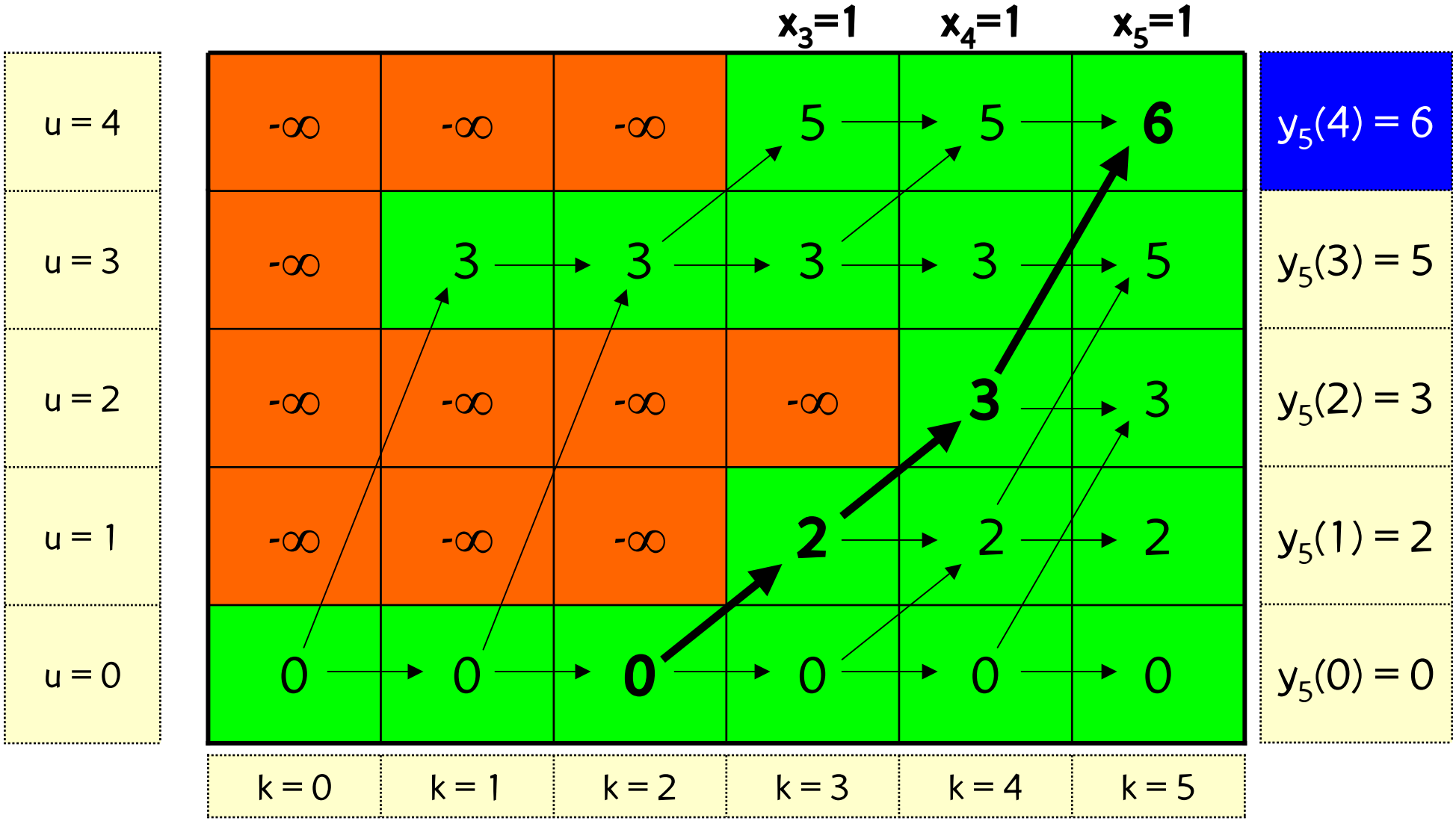
Problema da mochila



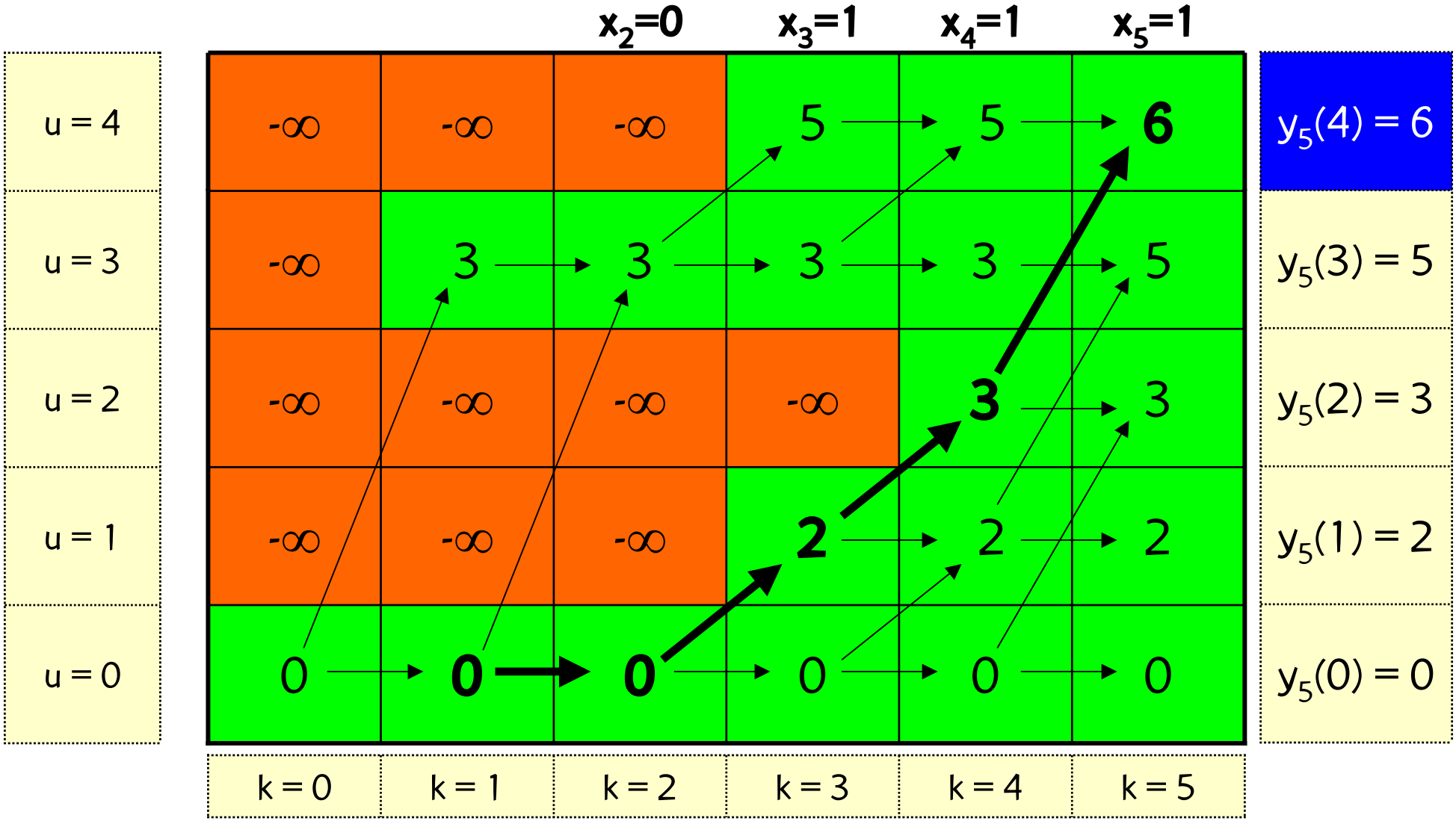
Problema da mochila



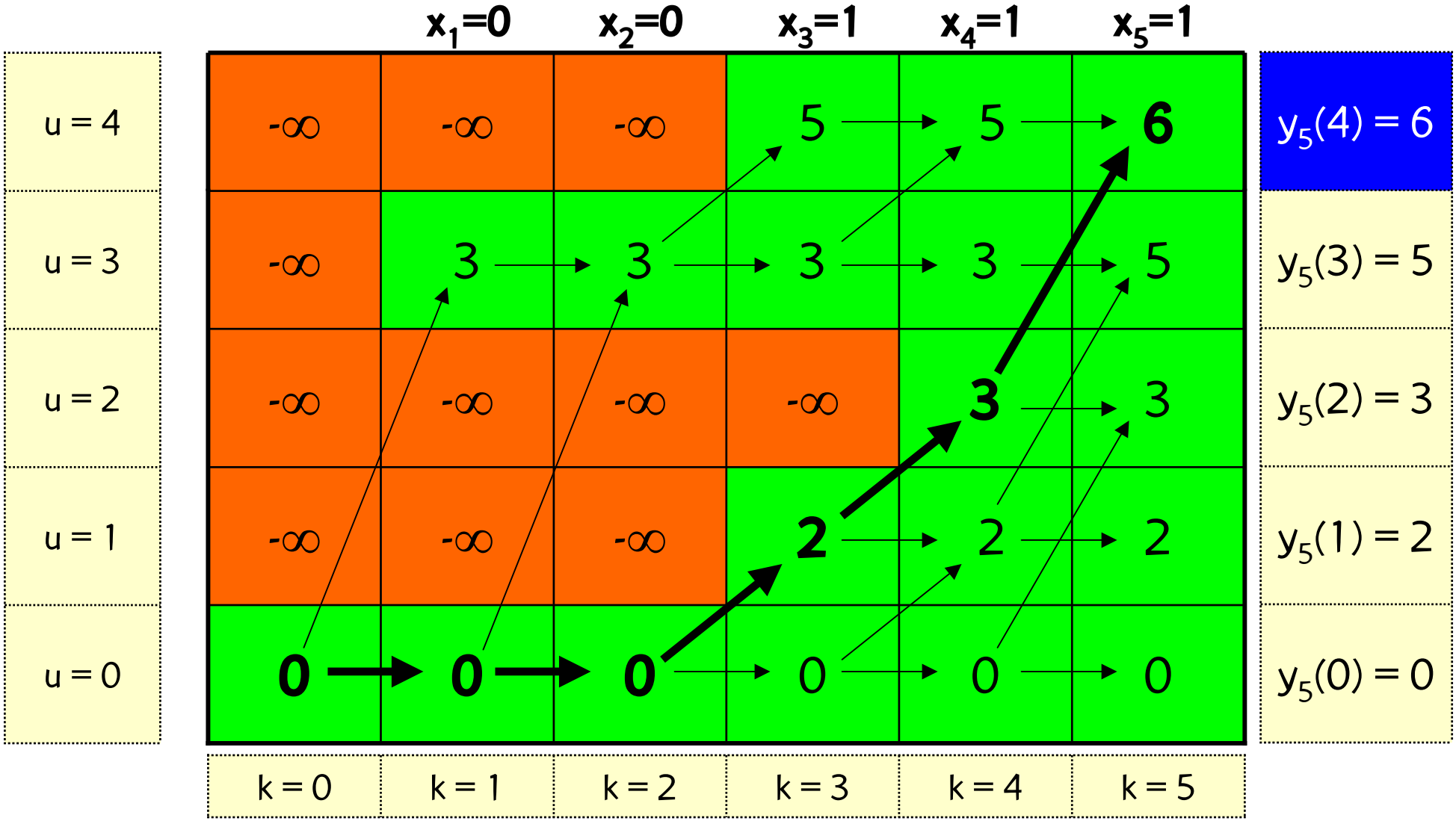
Problema da mochila



Problema da mochila



Problema da mochila

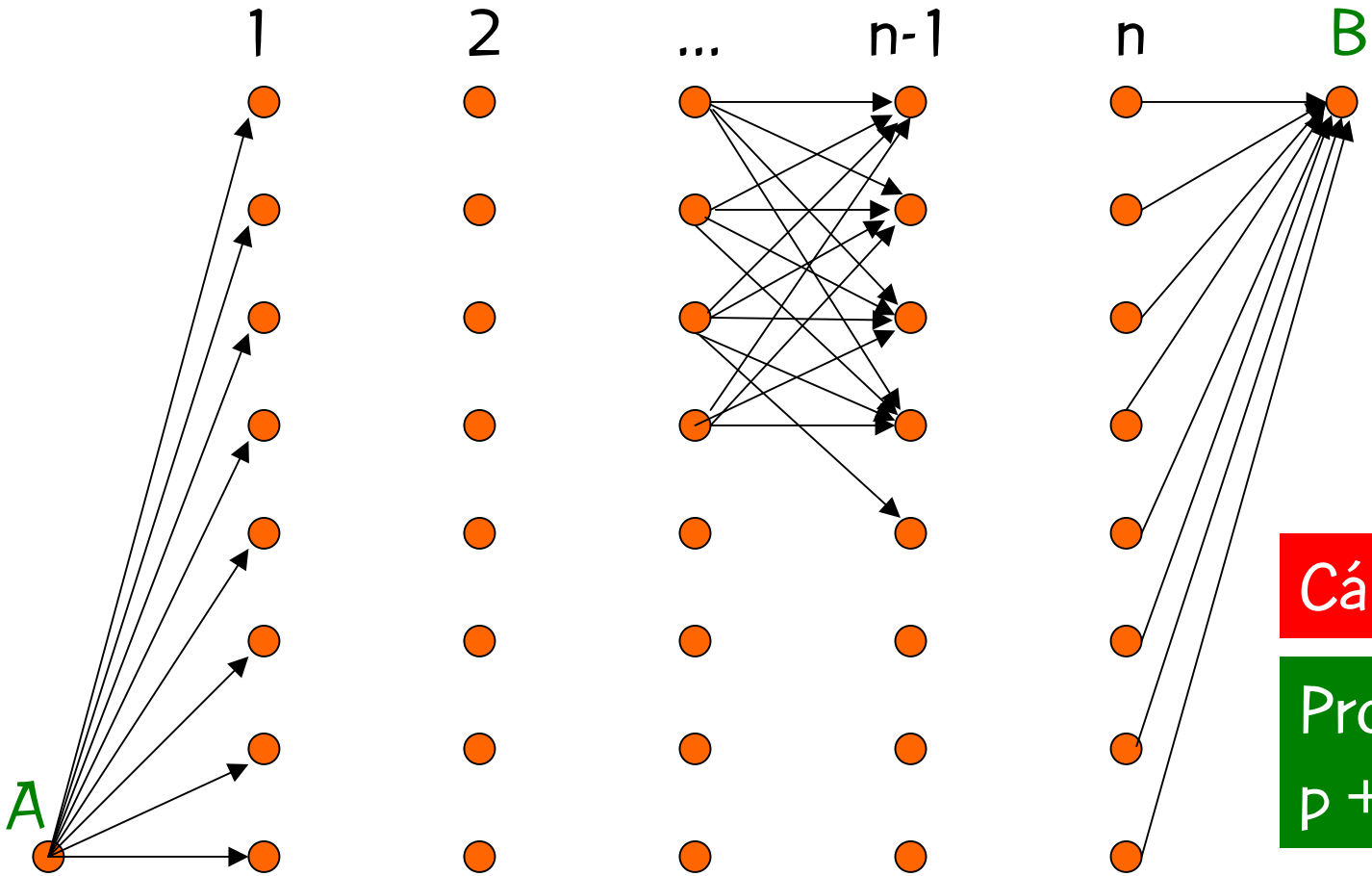


Problema da mochila

- Os estados em verde e as transições possíveis (arcos com setas) definem um grafo multiestágio para a aplicação da recursão de programação dinâmica.
- Número de operações (tempo de processamento) diretamente proporcional ao produto do tamanho da mochila pelo número de variáveis (preencher inteiramente a matriz de dimensões (b+1)x(n+1)): aplicabilidade limitada aos valores de n e de b
- Caso seja possível levar múltiplas cópias de cada item, aumenta o número de decisões e a complexidade do problema.

Programação dinâmica

- Redução no número total de operações:



p "linhas"

Cálculo da menor trajetória de A a B

Total de adições?

Total de trajetórias de A a B: p^n

Cálculo exaustivo: $n.p^n$

Programação dinâmica: $p + p.p.(n-1) + p \approx n.p^2$

Programação dinâmica

- Observar que o valor da função objetivo associado a um estado depende apenas dele, mas não depende da forma como foi atingido.
- O fato de serem implicitamente descartadas subtrajetórias não-ótimas é que leva à redução significativa do número de operações.

Programação dinâmica

- (I) Problema de otimização de investimentos: Considere-se um conjunto de n possíveis investimentos, nos quais é possível aplicar no máximo um total de M unidades monetárias. Seja $c_j(k)$ o retorno obtido com a aplicação de $0 \leq k \leq M$ unidades monetárias no investimento $j = 1,\ldots,n$. Determinar a aplicação ótima a ser feita em cada investimento, de modo a maximizar o retorno total. Resolver o problema para $n = 3$ e $M = 4$, considerando a matriz de retornos fornecida a seguir:

Programação dinâmica

k	$c_1(k)$	$c_2(k)$	$c_3(k)$
0	0	0	0
1	3	1	2
2	7	2	4
3	11	4	6
4	12	8	8

Programação dinâmica

- Estágio:
  - cada investimento k = 1, 2, 3 considerado
- Estado:
  - total ainda disponível para ser aplicado nos investimentos k, k+1, ..., n
  - outra alternativa: total aplicado nos investimentos 1, 2, ..., k
- Decisão:
  - quanto aplicar no investimento k
- Recursão:
  - $R_k(x) = \max_{0 \le y \le x} \{c_k(y) + R_{k+1}(x-y)\}$

Programação dinâmica

- Condições de contorno:
  - $R_4(0) = 0$

    Todos os recursos devem ser usados nos três investimentos.
- Valor máximo do retorno obtido:
  - $R_1(4) = ?$

    Maximizar o retorno obtido com a aplicação de um total de quatro unidades nos investimentos 1, 2, 3.

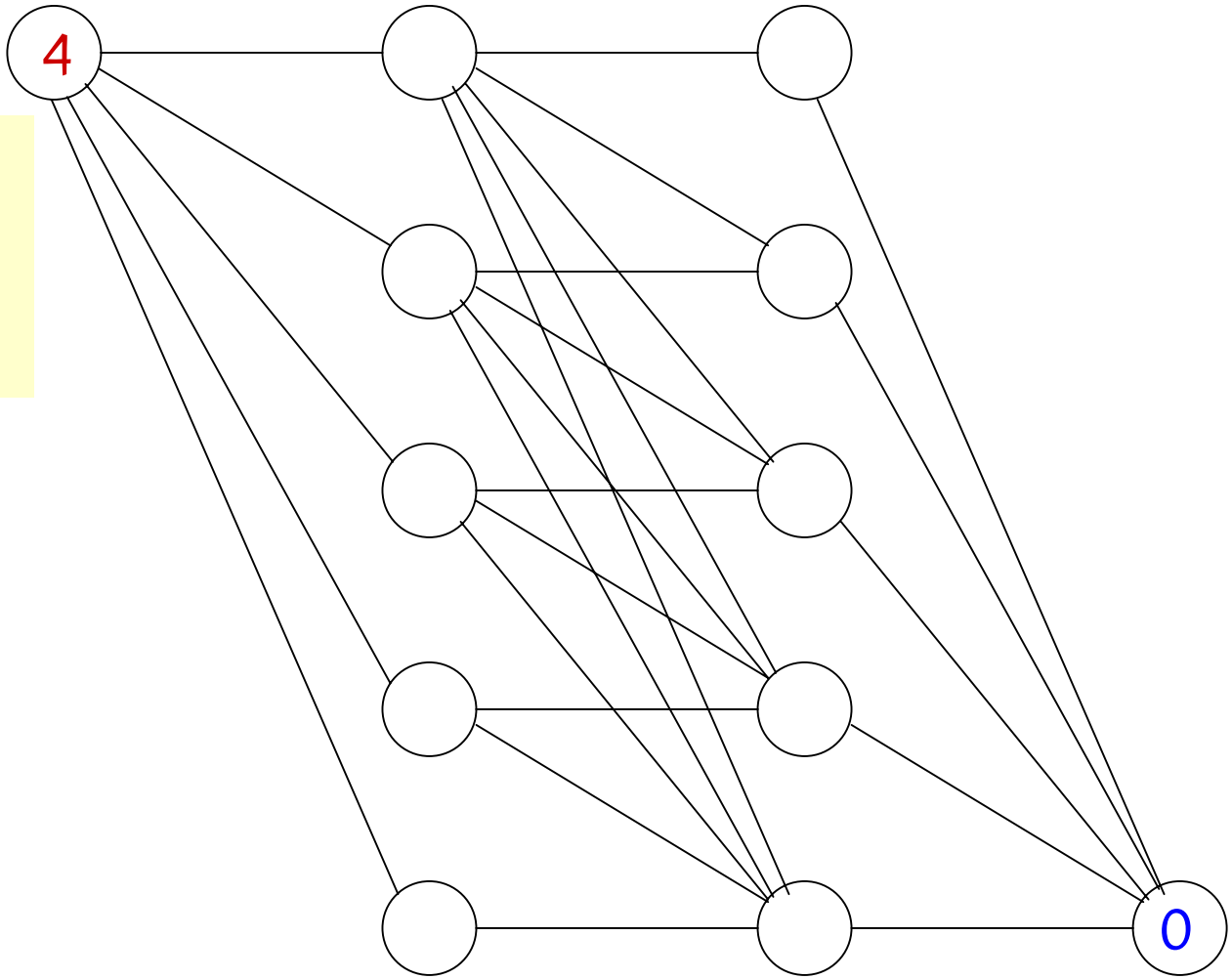
Programação dinâmica

Estado inicial:
$R_1(4) = ?$

Estado final:
$R_4(0) = 0$

Investimento 1 Investimento 2 Investimento 3

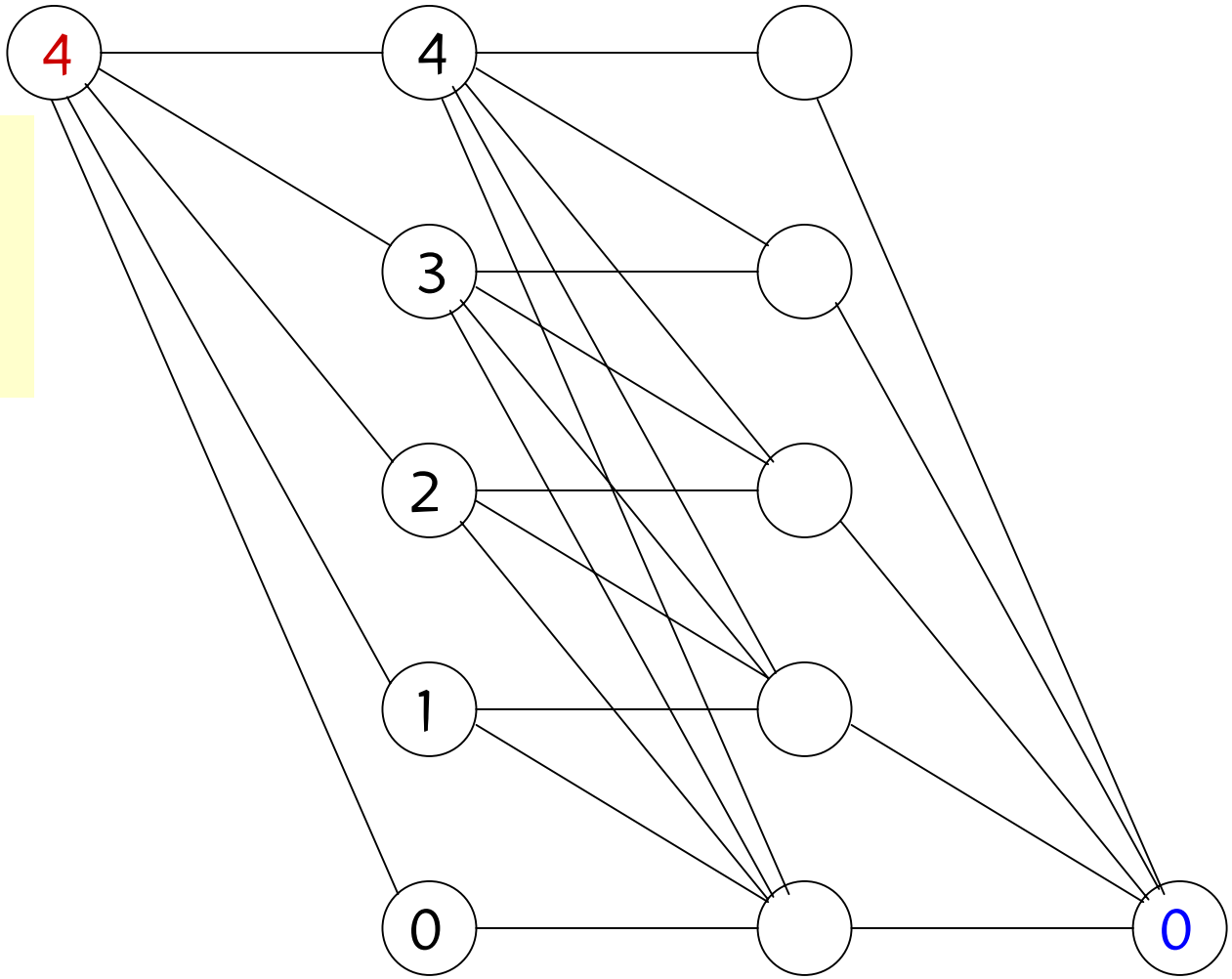
Programação dinâmica



Estado inicial:
$R_1(4) = ?$

Estado final:
$R_4(0) = 0$

Programação dinâmica

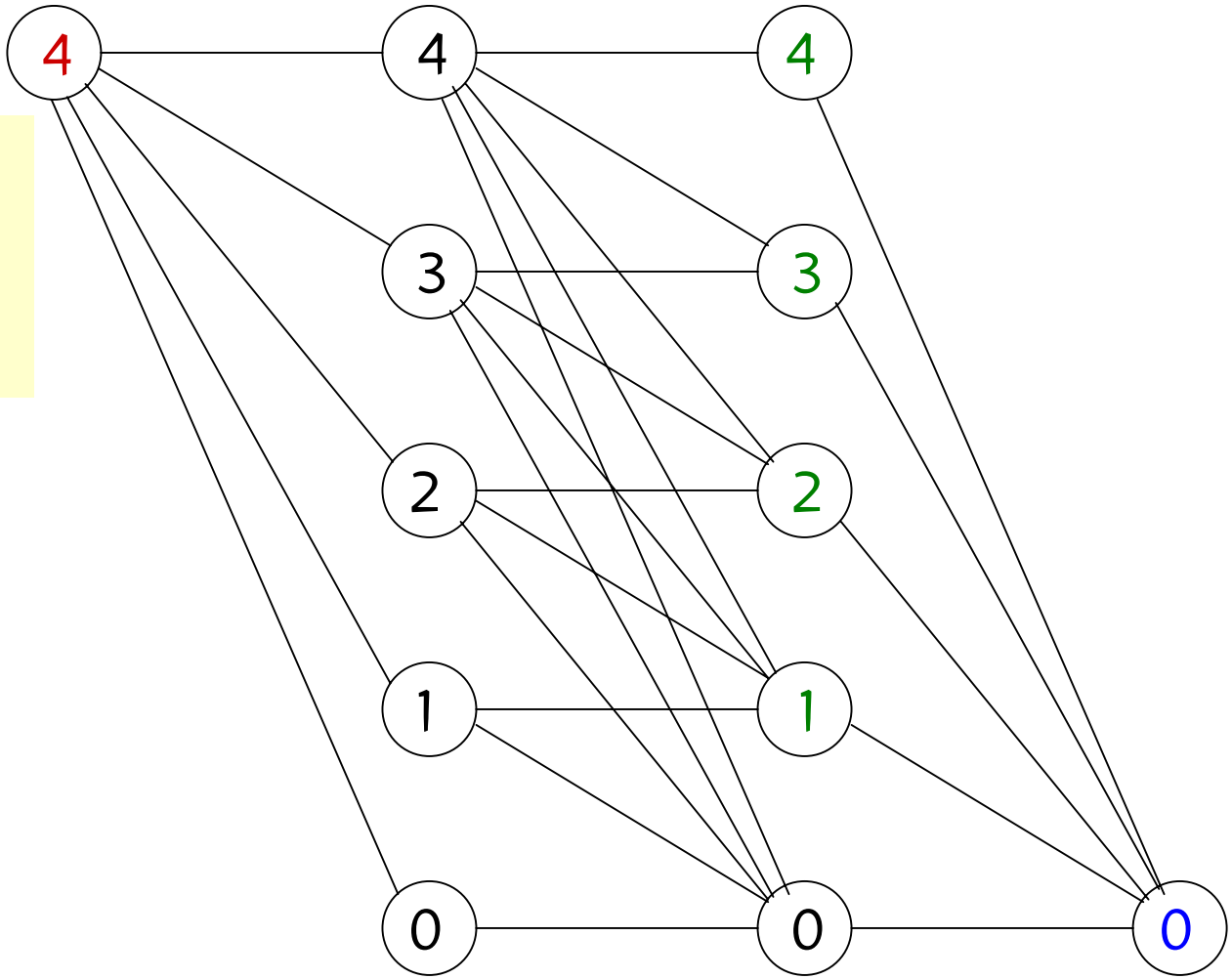


Estado inicial:
$R_1(4) = ?$

Estado final:
$R_4(0) = 0$

Programação dinâmica

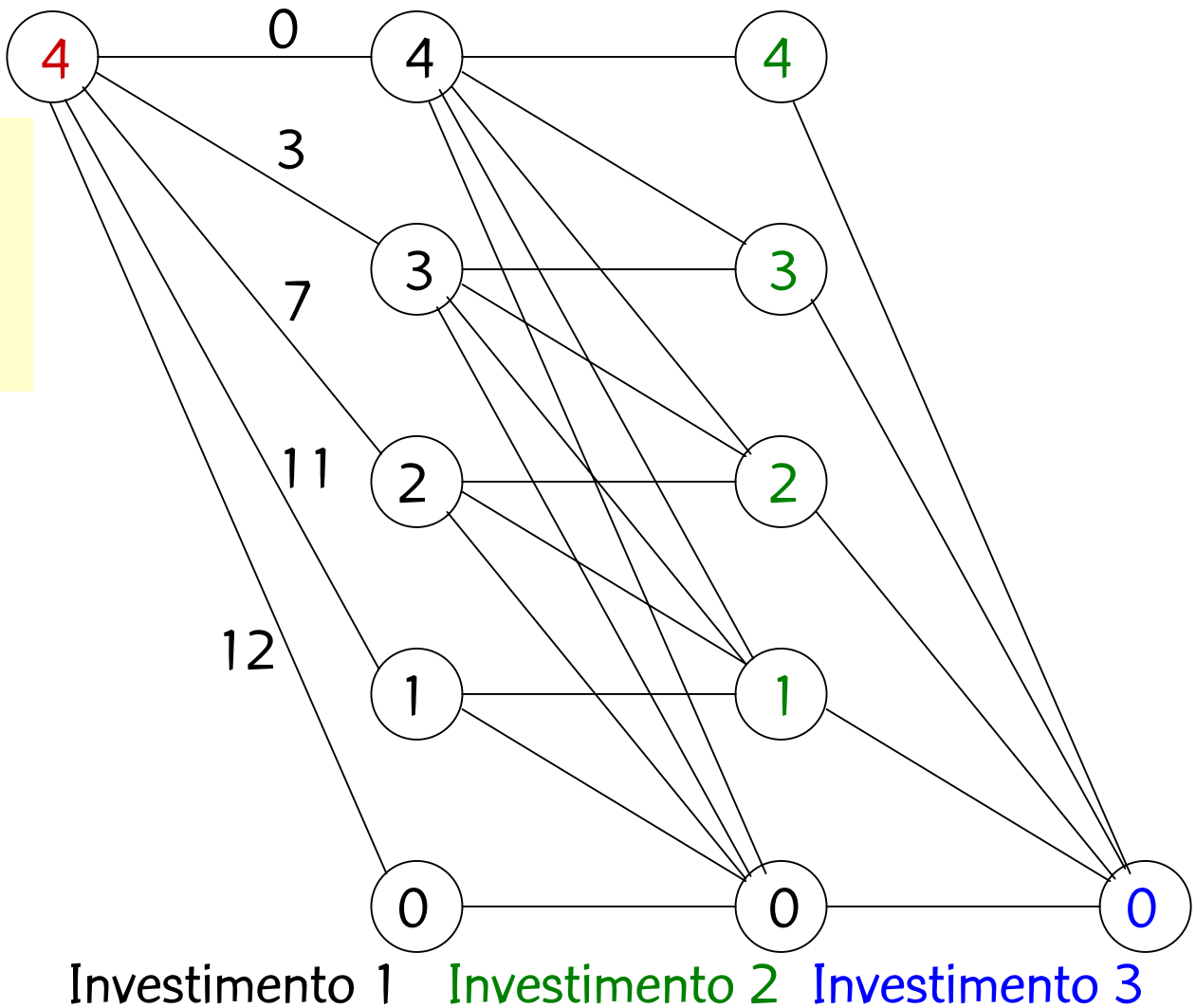
Estado inicial: $R_1(4) = ?$



Estado final: $R_4(0) = 0$

Programação dinâmica

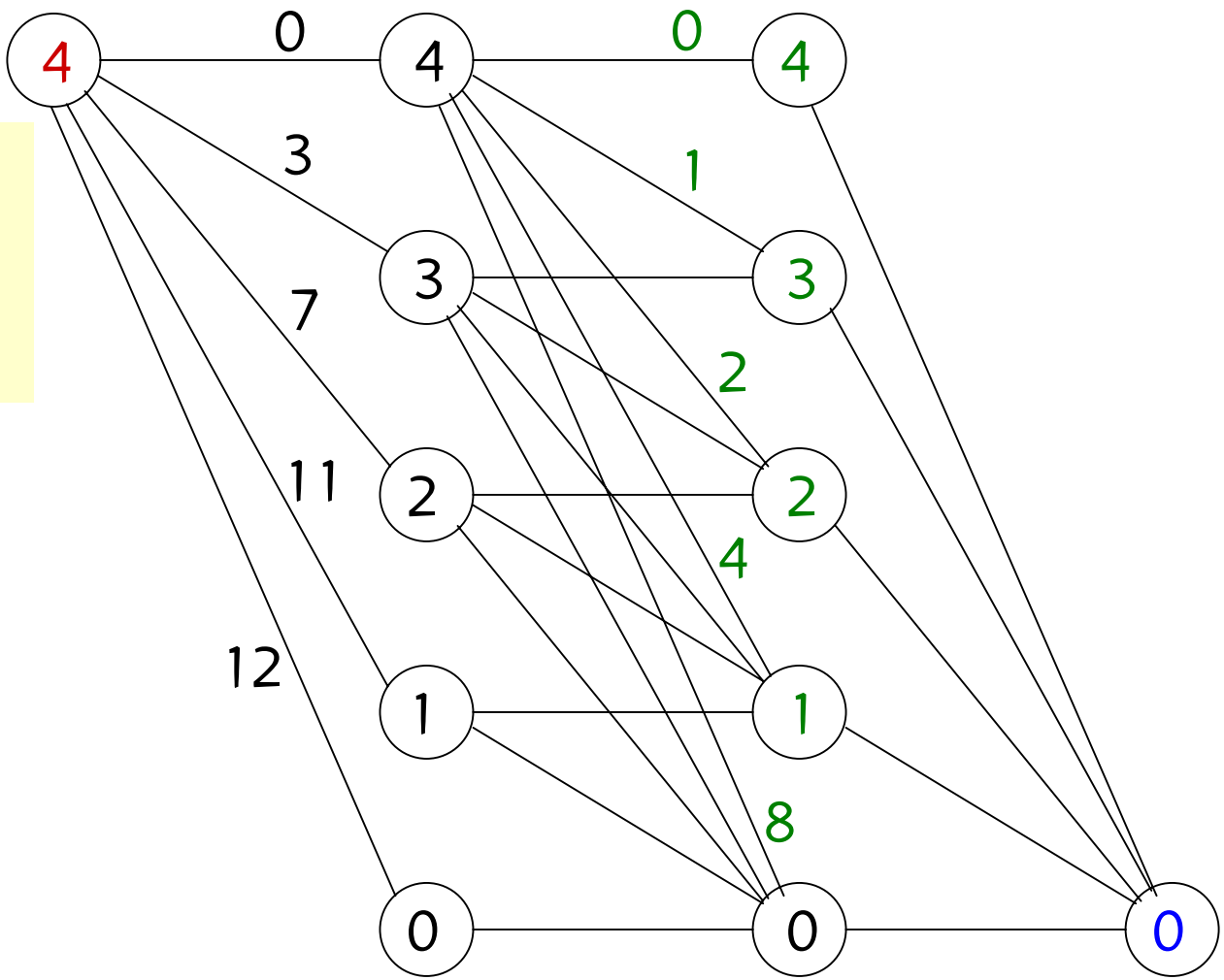
Estado inicial: $R_1(4) = ?$



Estado final: $R_4(0) = 0$

Programação dinâmica

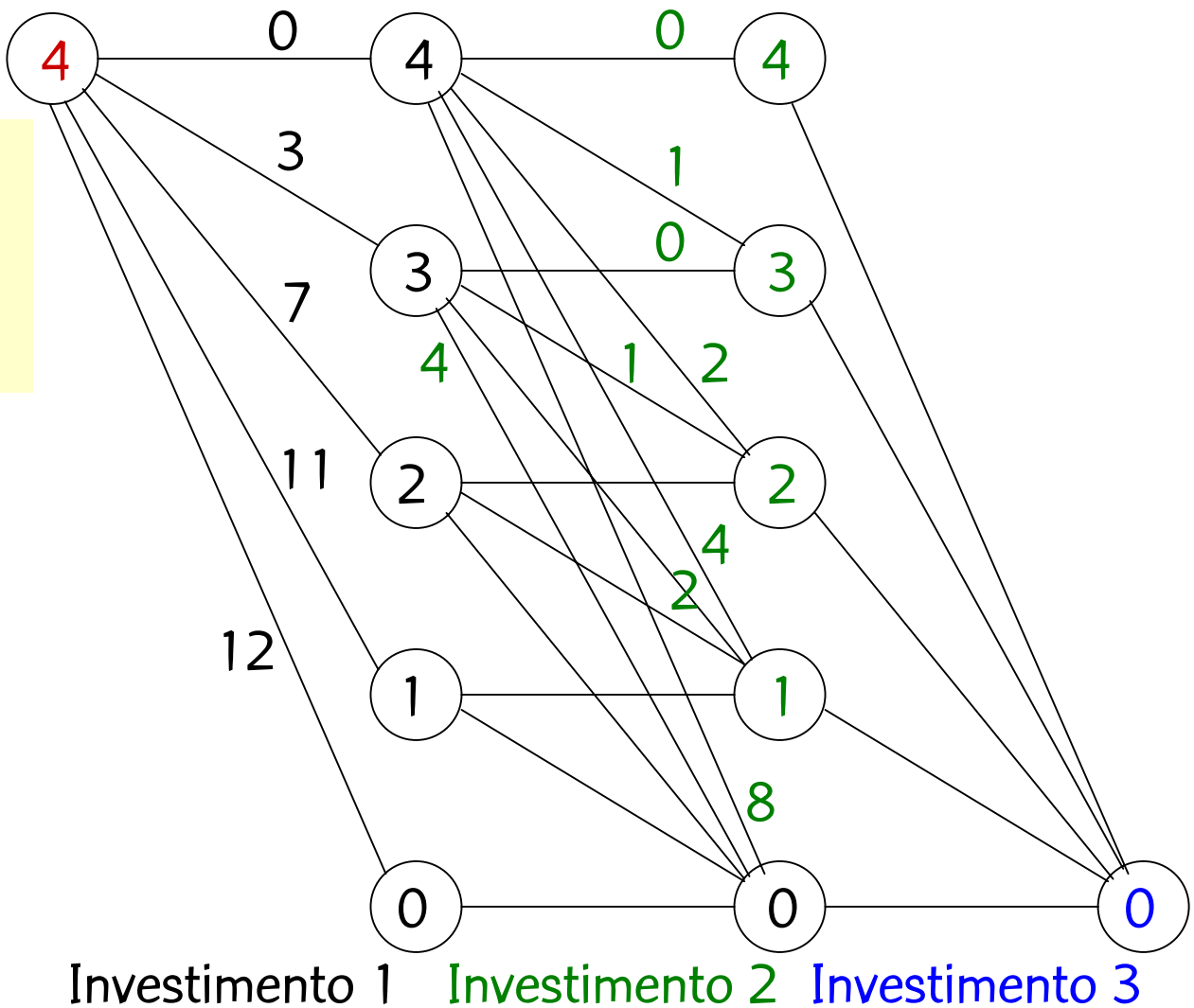
Estado inicial: $R_1(4) = ?$



Estado final: $R_4(0) = 0$

Programação dinâmica

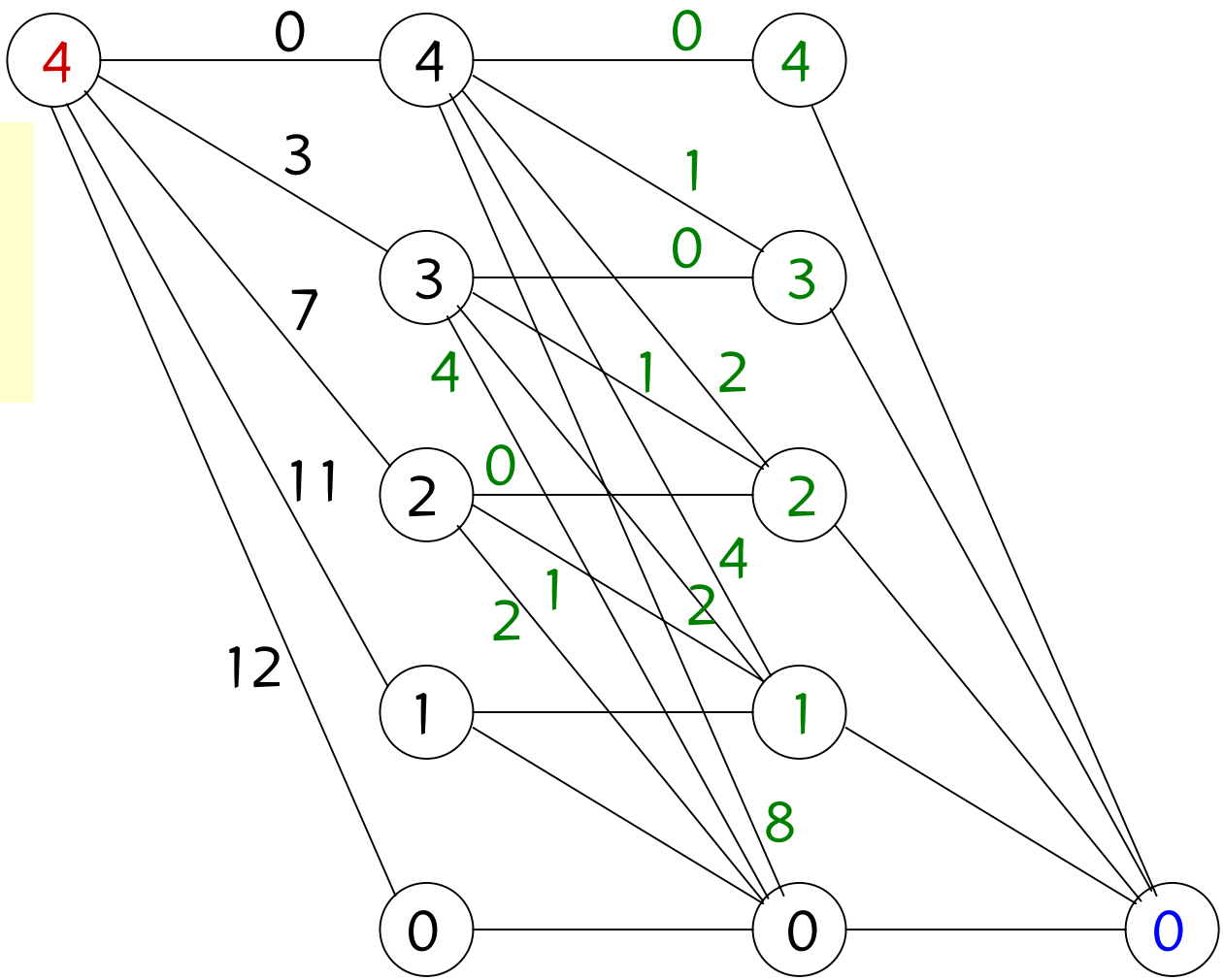
Estado inicial: $R_1(4) = ?$



Estado final: $R_4(0) = 0$

Programação dinâmica

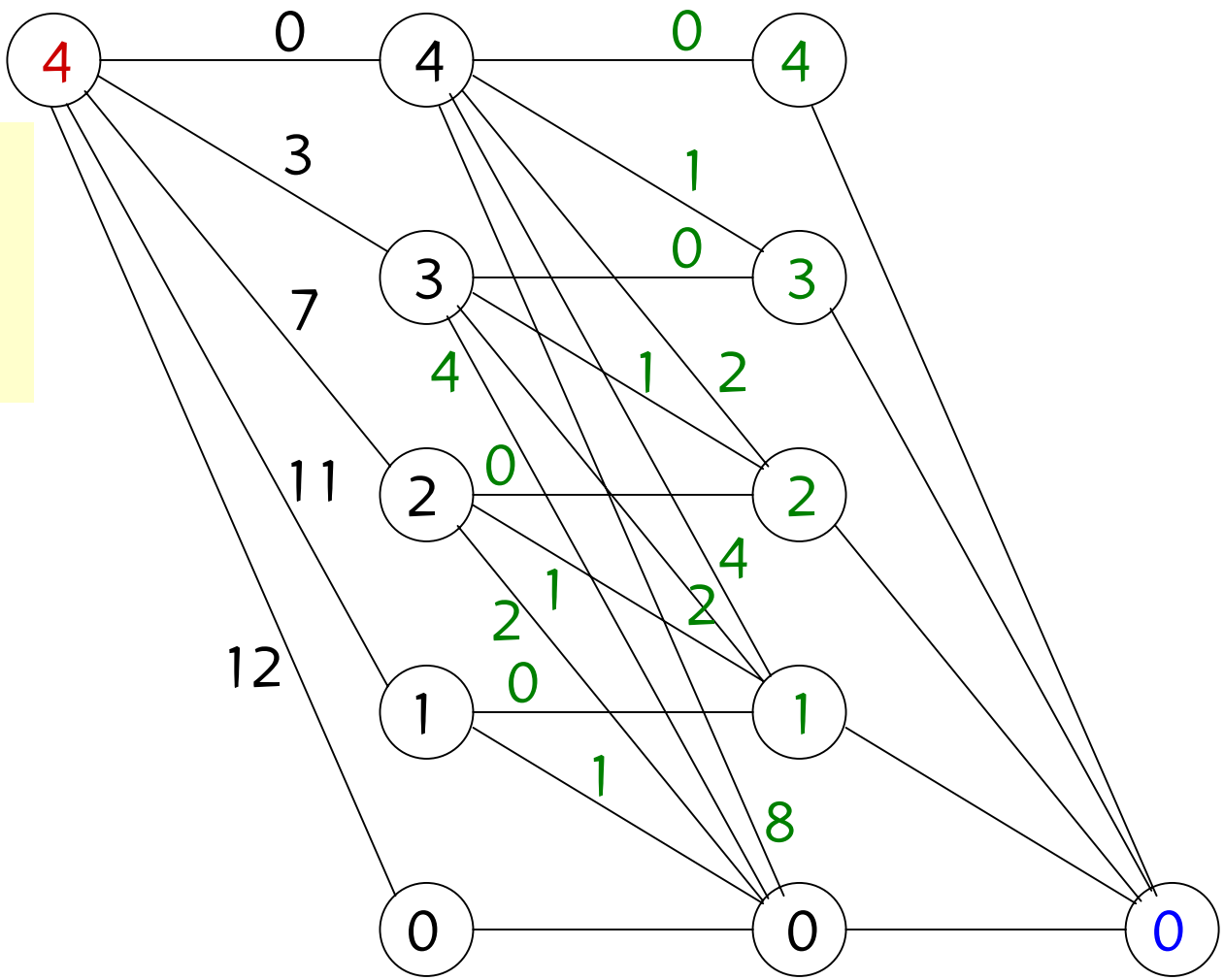
Estado inicial: $R_1(4) = ?$



Estado final: $R_4(0) = 0$

Programação dinâmica

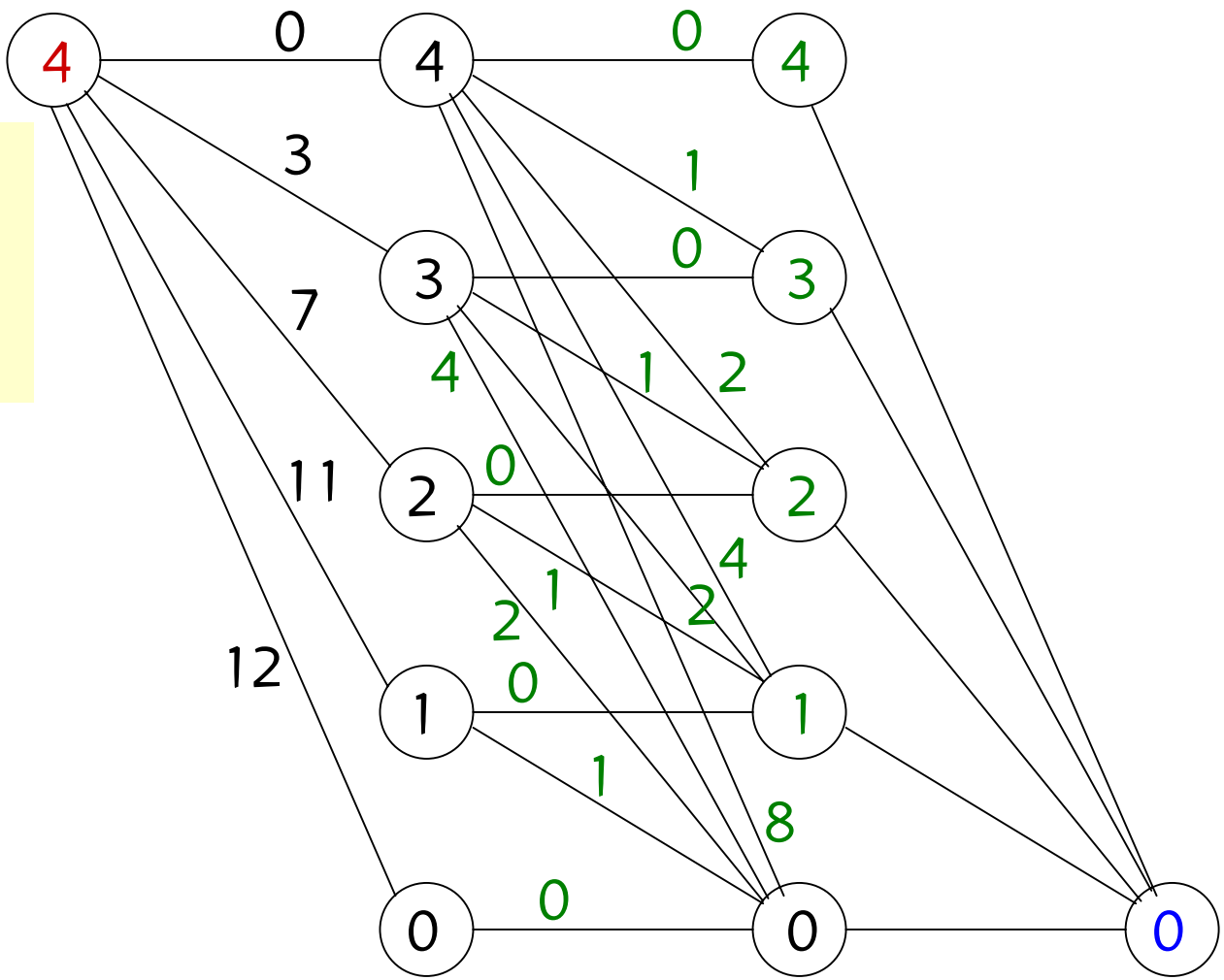
Estado inicial: $R_1(4) = ?$



Estado final: $R_4(0) = 0$

Programação dinâmica

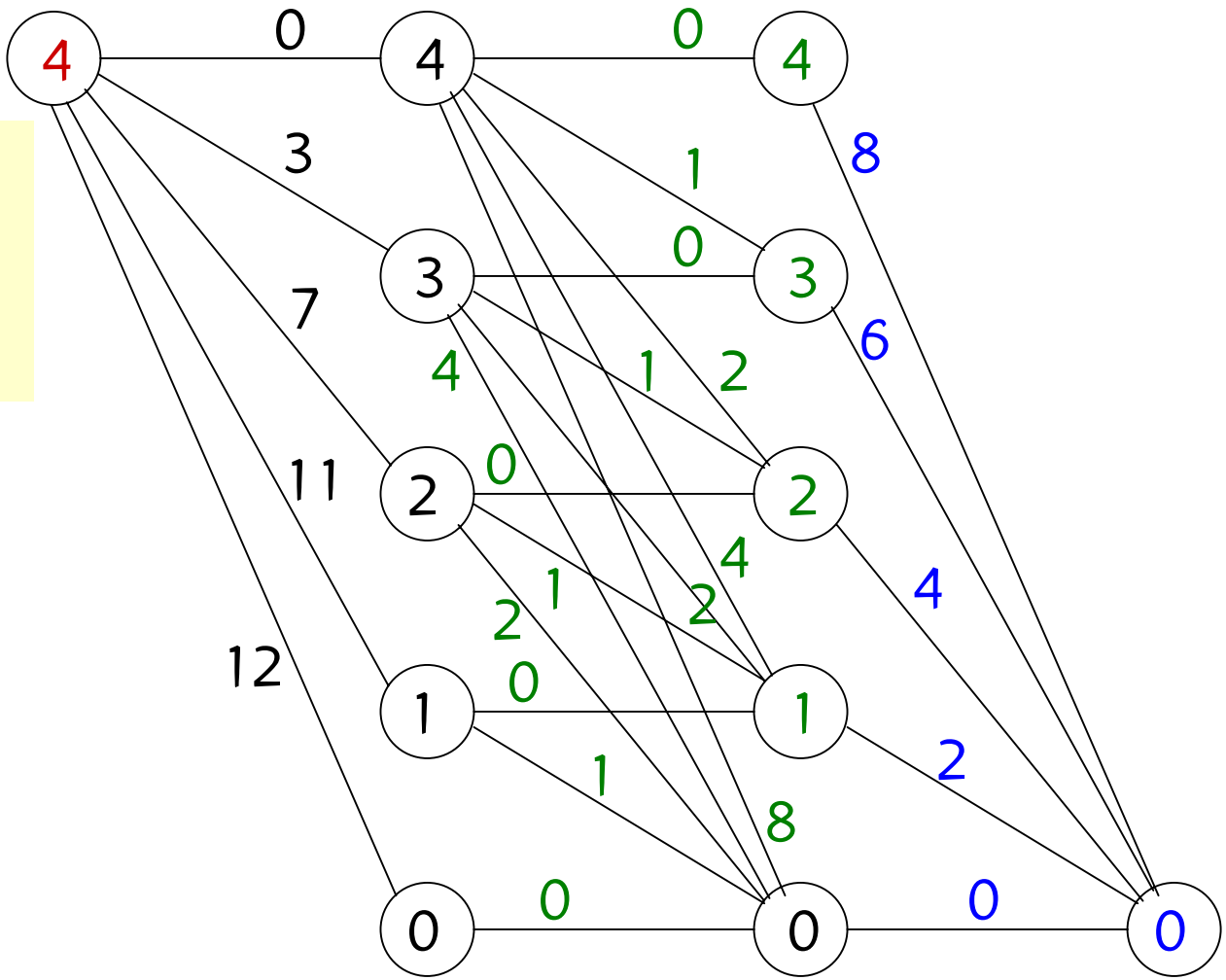
Estado inicial:
$R_1(4) = ?$



Estado final:
$R_4(0) = 0$

Programação dinâmica

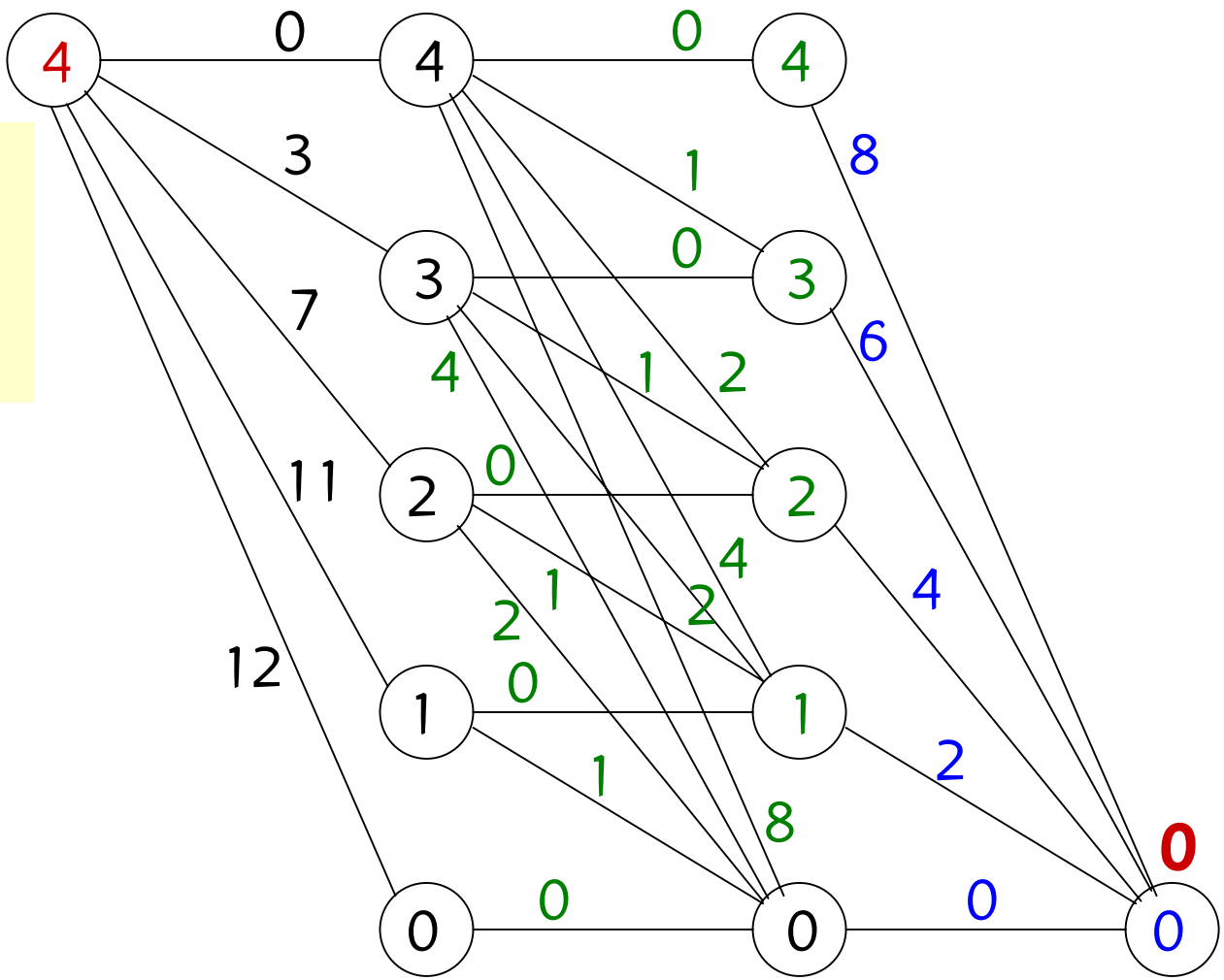
Estado inicial:
$R_1(4) = ?$



Estado final:
$R_4(0) = 0$

Programação dinâmica

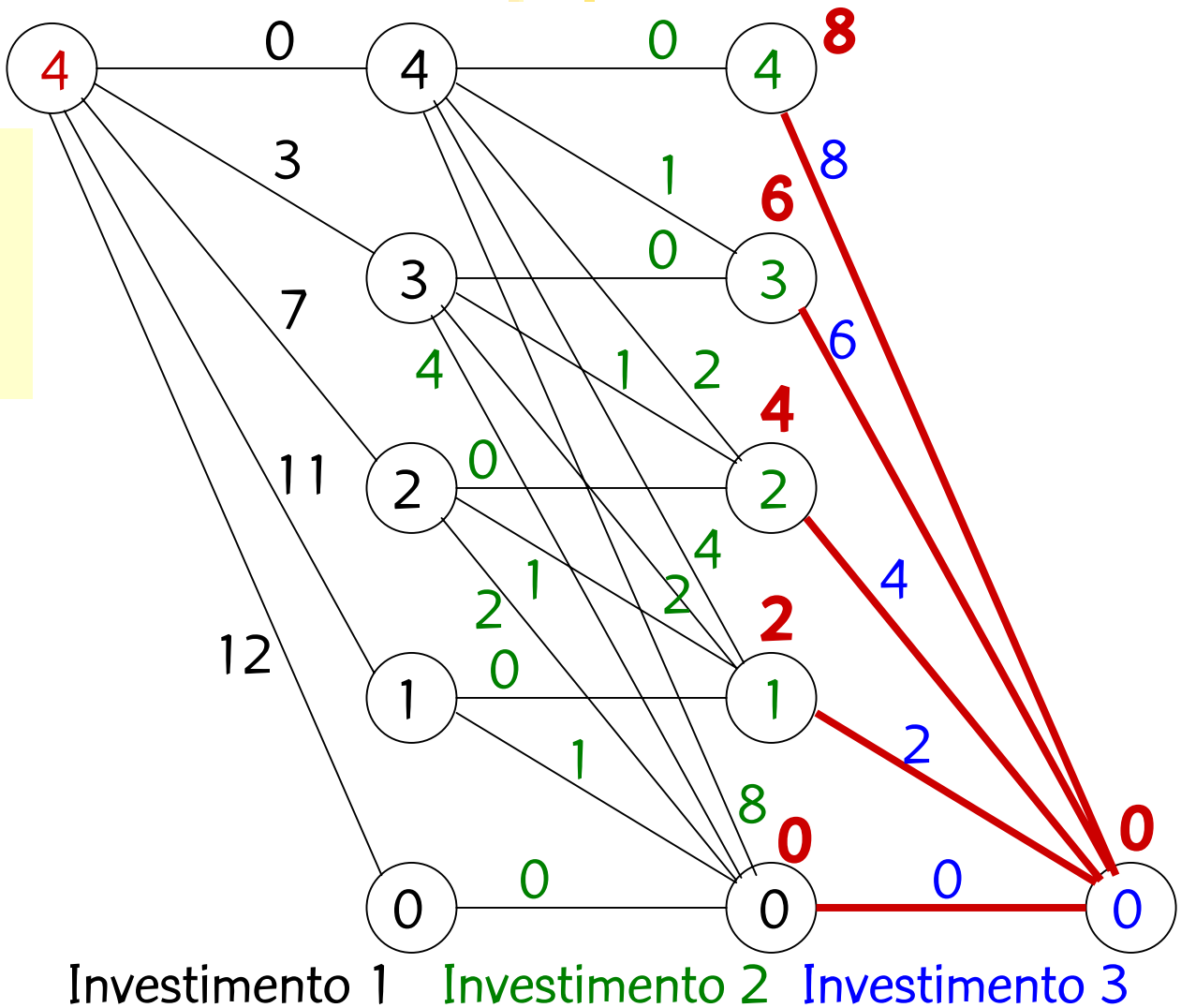
Estado inicial: $R_1(4) = ?$



Estado final: $R_4(0) = 0$

Programação dinâmica

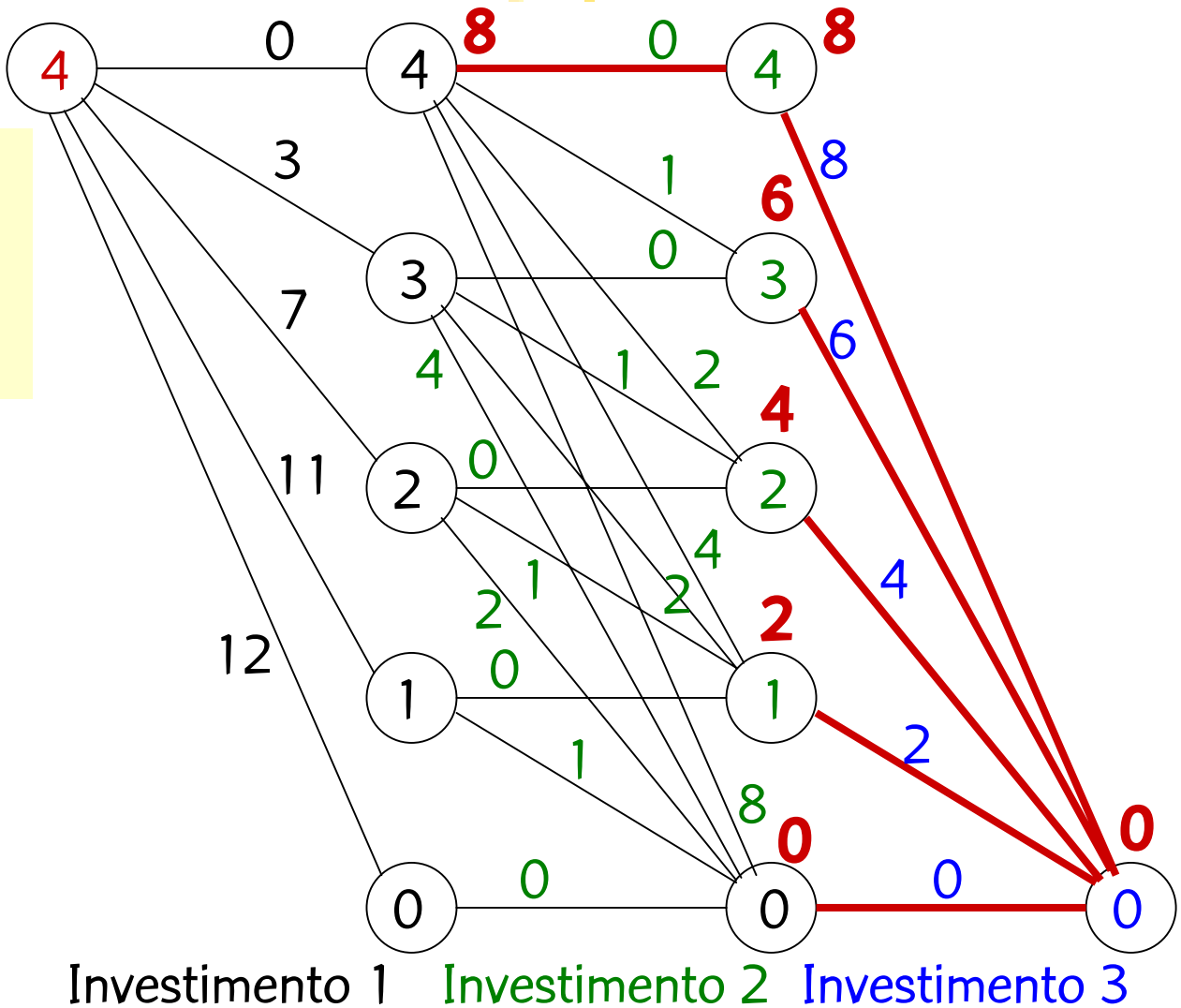
Estado inicial:
$R_1(4) = ?$



Estado final:
$R_4(0) = 0$

Programação dinâmica

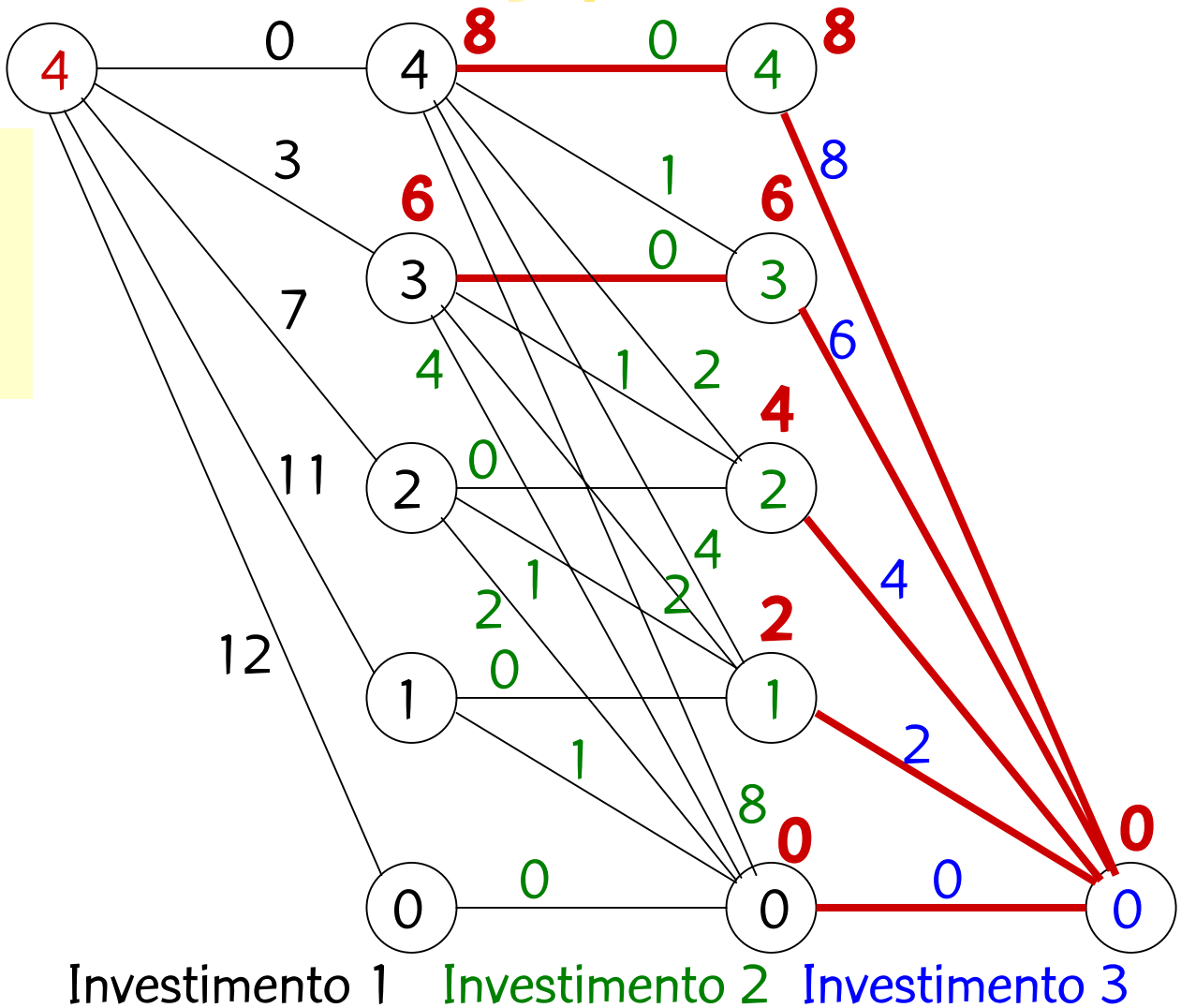
Estado inicial:
$R_1(4) = ?$



Estado final:
$R_4(0) = 0$

Programação dinâmica

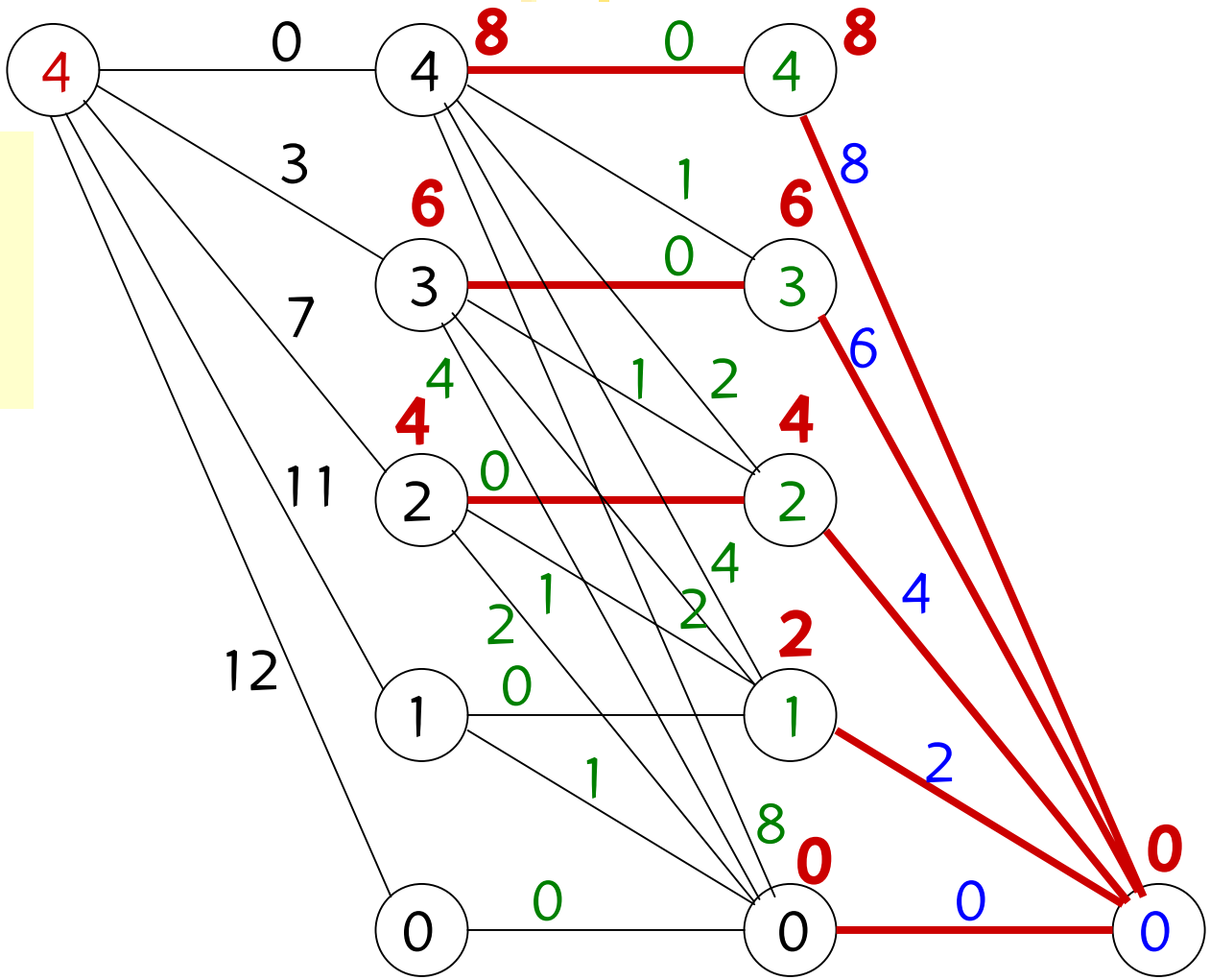
Estado inicial:
$R_1(4) = ?$



Estado final:
$R_4(0) = 0$

Programação dinâmica

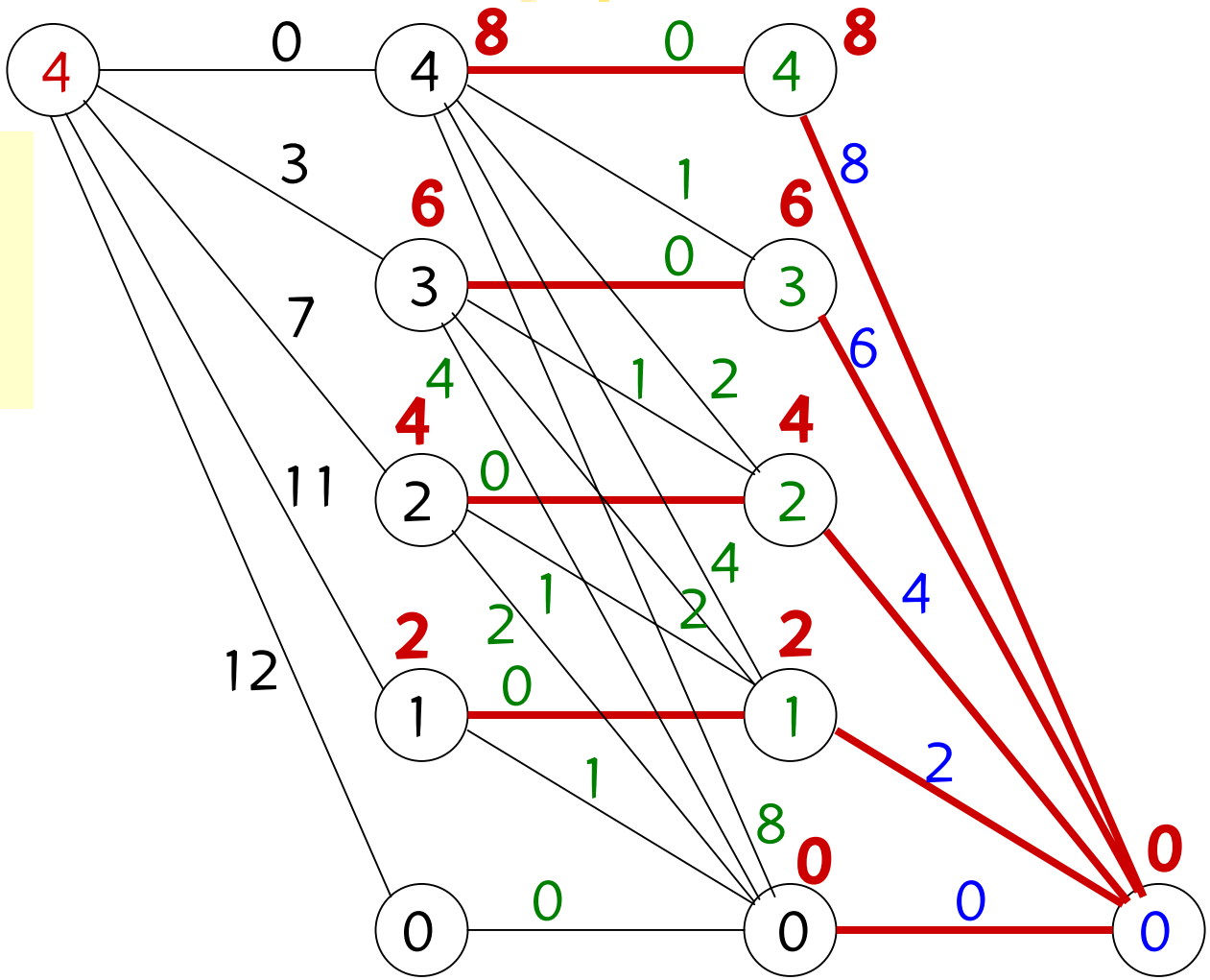
Estado inicial:
$R_1(4) = ?$



Estado final:
$R_4(0) = 0$

Programação dinâmica

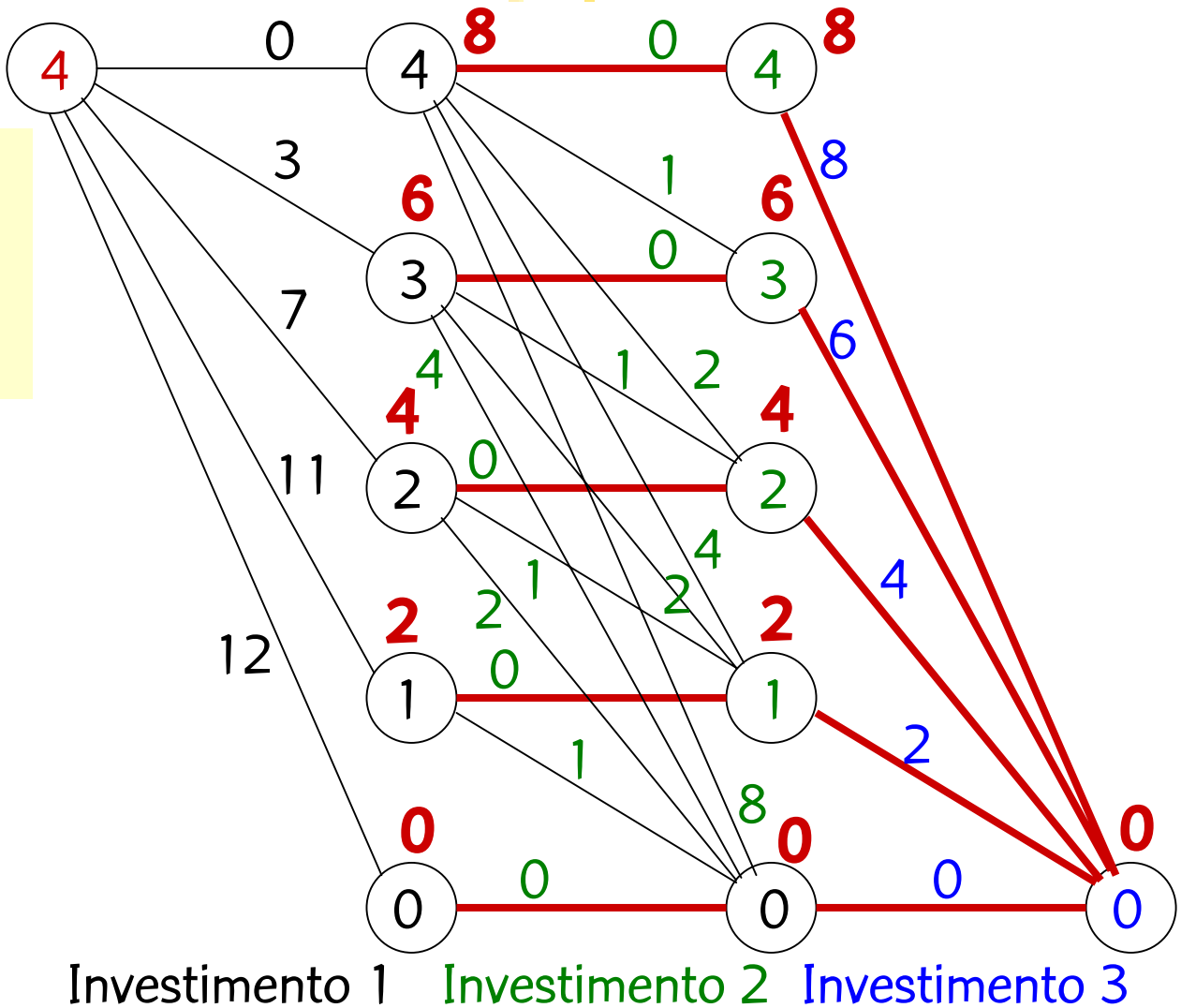
Estado inicial: $R_1(4) = ?$



Estado final: $R_4(0) = 0$

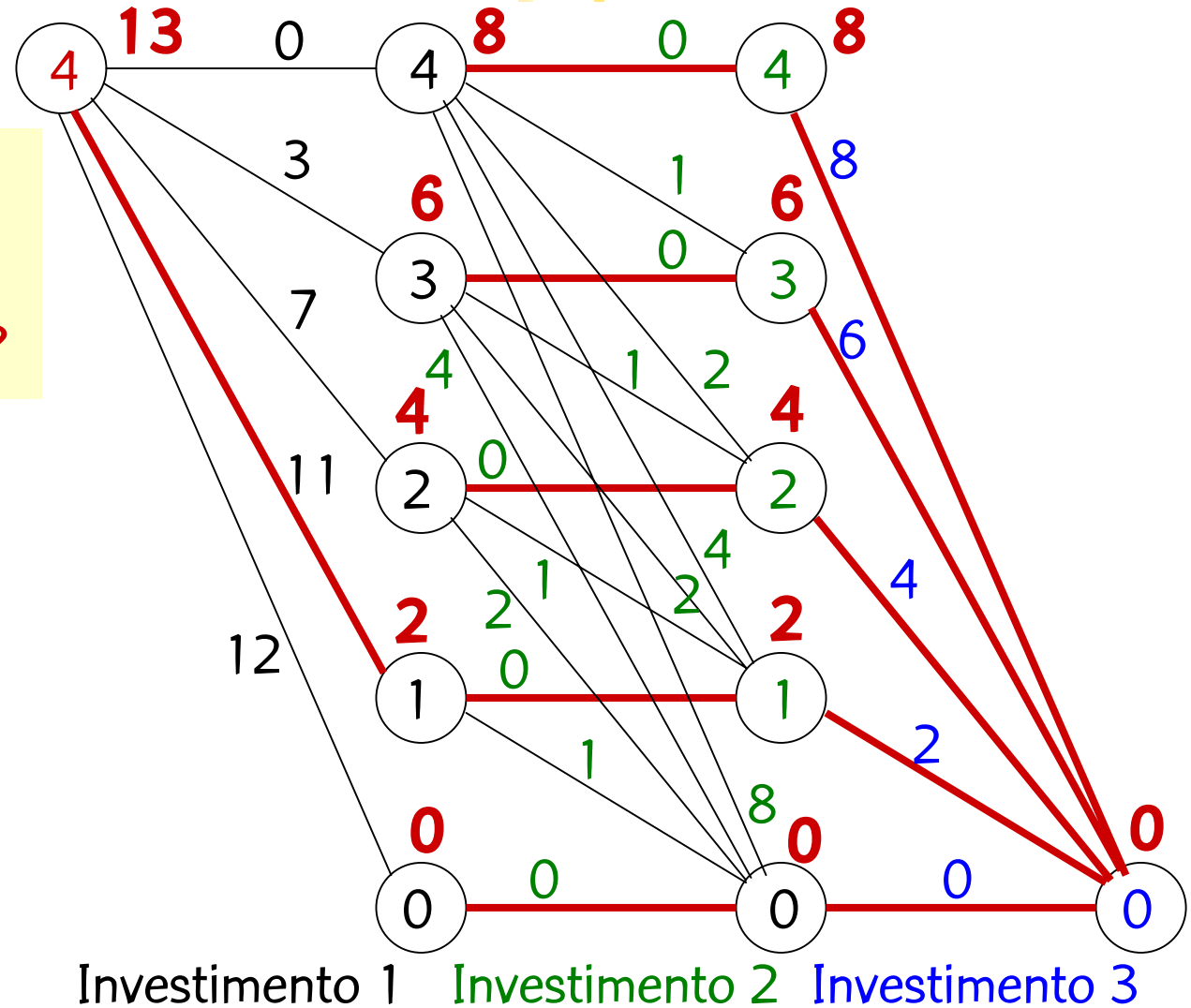
Programação dinâmica

Estado inicial:
$R_1(4) = ?$



Estado final:
$R_4(0) = 0$

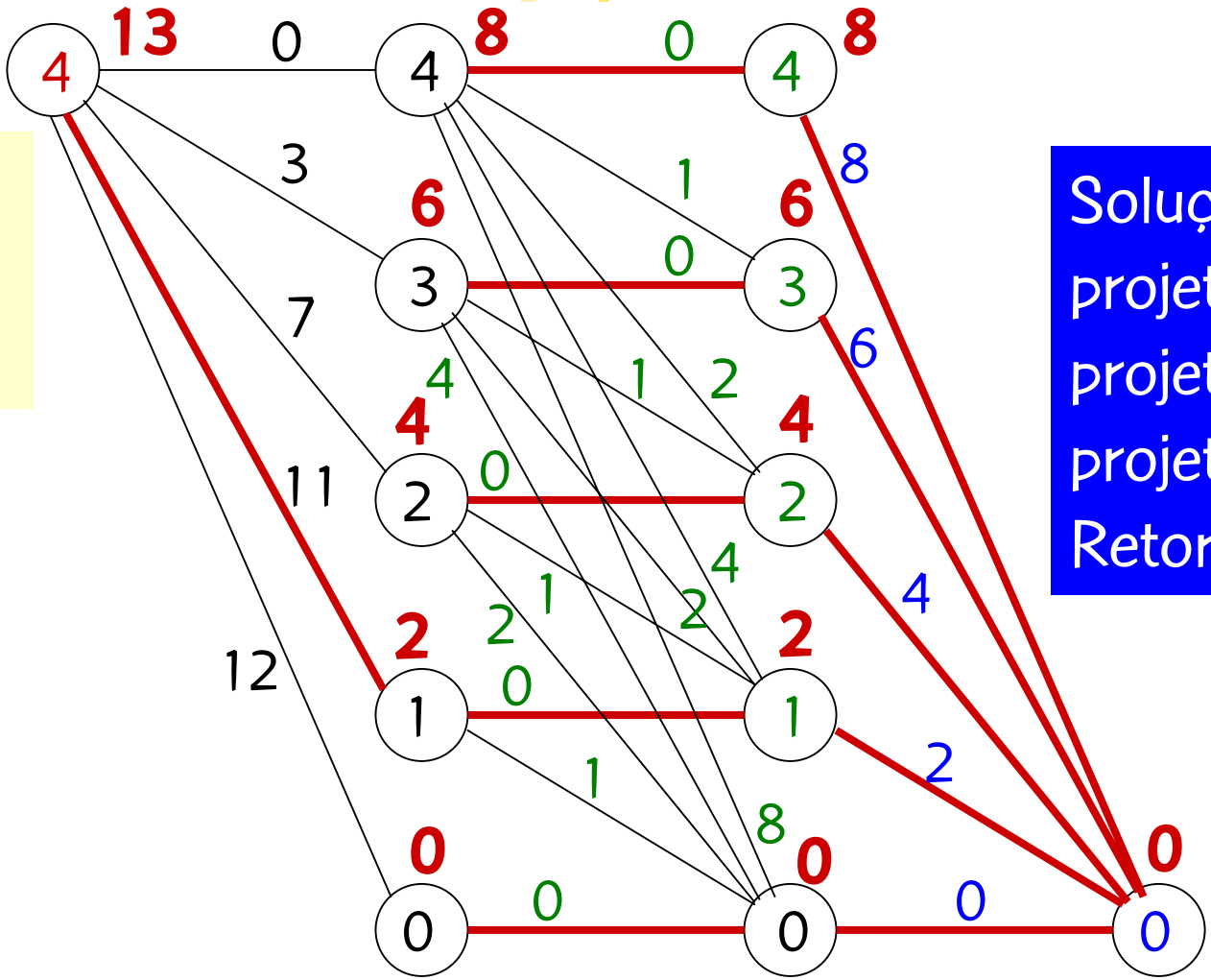
Programação dinâmica



Estado inicial: $R_1(4) = ?$

Estado final: $R_4(0) = 0$

Programação dinâmica



Estado inicial:
$R_1(4) = ?$

Solução ótima:
projeto 1: 3 unidades
projeto 2: 0 unidades
projeto 3: 1 unidade
Retorno total = 13

Estado final:
$R_4(0) = 0$

Programação dinâmica

- (II) Problema de substituição de equipamentos: Uma empresa precisa de uma máquina para produzir determinado produto ao longo dos próximos N anos. Os seguintes dados são disponíveis:

  P = preço de compra de uma máquina nova

  c(i) = custo anual de operação de uma máquina de idade i

  t(i) = preço obtido na troca de uma máquina de idade i

  r(i) = valor de residual uma máquina de idade i

  No início do período de programação, a empresa dispõe

Programação dinâmica

de uma máquina com M anos de uso. Determinar a política ótima de substituição desta máquina ao longo dos N anos, de modo a minimizar a soma dos custos anuais. Considerar N = 5 (horizonte), M = 2 (idade da máquina no início do horizonte), P = 50 (preço de compra de uma máquina nova) e os demais custos e preços informados na tabela a seguir.

- Observar que o modelo pode ser estendido para considerar preços diferentes para uma máquina nova a cada ano, taxa de desconto, troca por máquinas mais novas mas já usadas, etc.

Programação dinâmica

idade i	0	1	2	3	4	5	6
c(i)	10	13	20	40	70	100	110
t(i)	-	32	21	11	5	0	0
r(i)	-	25	17	8	0	0	0

Programação dinâmica

- Estágio:
  - cada ano k = 1, 2, ..., N do horizonte
- Estado:
  - idade da máquina no ano k
- Decisão:
  - o que fazer com a máquina no ano k: trocá-la por uma nova ou mantê-la em funcionamento
- Recursão:
  - $V_k(i)$: custo ótimo total do ano k até o final do horizonte a partir de uma máquina de idade i

Programação dinâmica

- Recursão:
  - $V_k(i)$: custo ótimo total do ano k até o final do horizonte a partir de uma máquina de idade i
  - Máquinas novas são sempre compradas no dia 1 de janeiro de cada ano e completam um ano de vida em 1 de janeiro do ano seguinte.
  - Caso uma máquina de idade i seja trocada por uma nova no início do ano k: $V_k(i) = p - t(i) + c(0) + V_{k+1}(1)$
  - Caso contrário: $V_k(i) = c(i) + V_{k+1}(i+1)$
  - $V_k(i) = \text{mínimo } \{p - t(i) + c(0) + V_{k+1}(1), c(i) + V_{k+1}(i+1)\}$

Programação dinâmica

- Condições de contorno:
  - $V_{N+1}(i) = -r(i)$
    A máquina é vendida por seu valor residual ao final do horizonte.
- Início do ano 1: máquina tem idade M
- Custo mínimo ao longo do horizonte:
  - $V_1(M) = ?$

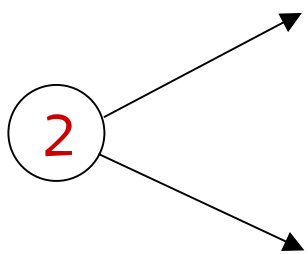
Programação dinâmica

Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

Programação dinâmica

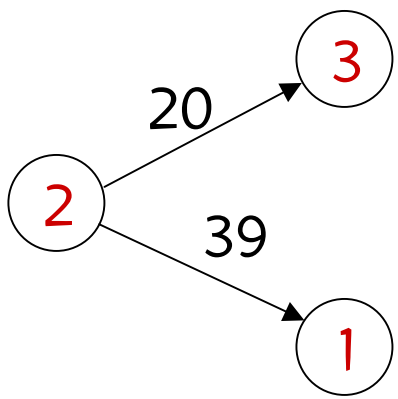
Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

Programação dinâmica



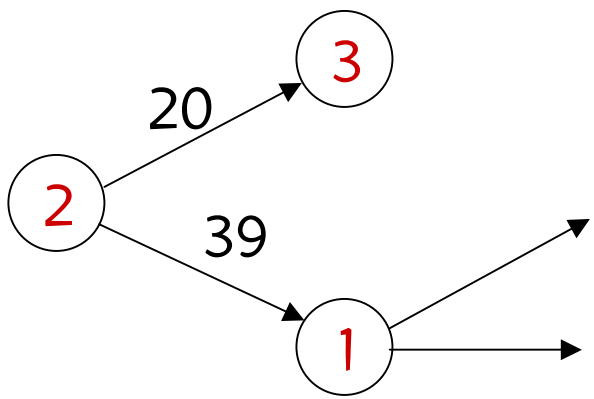
Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

Programação dinâmica



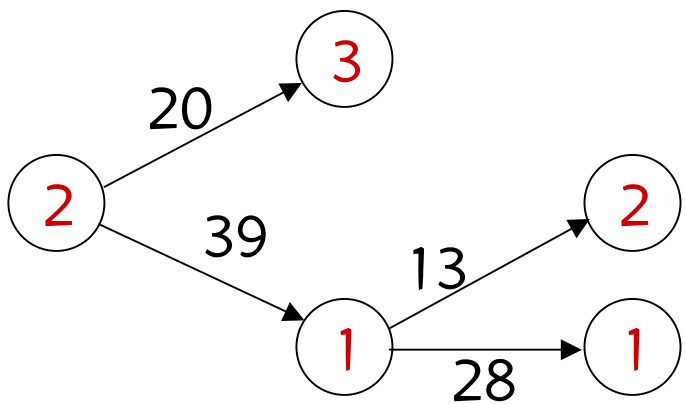
Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

Programação dinâmica



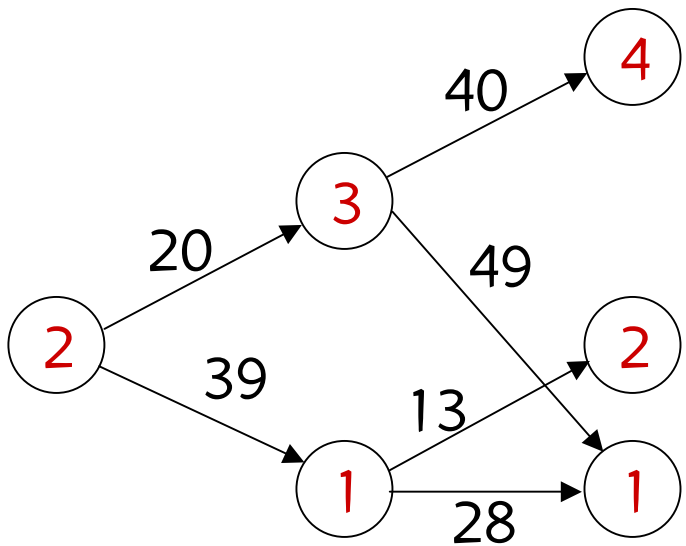
Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

Programação dinâmica



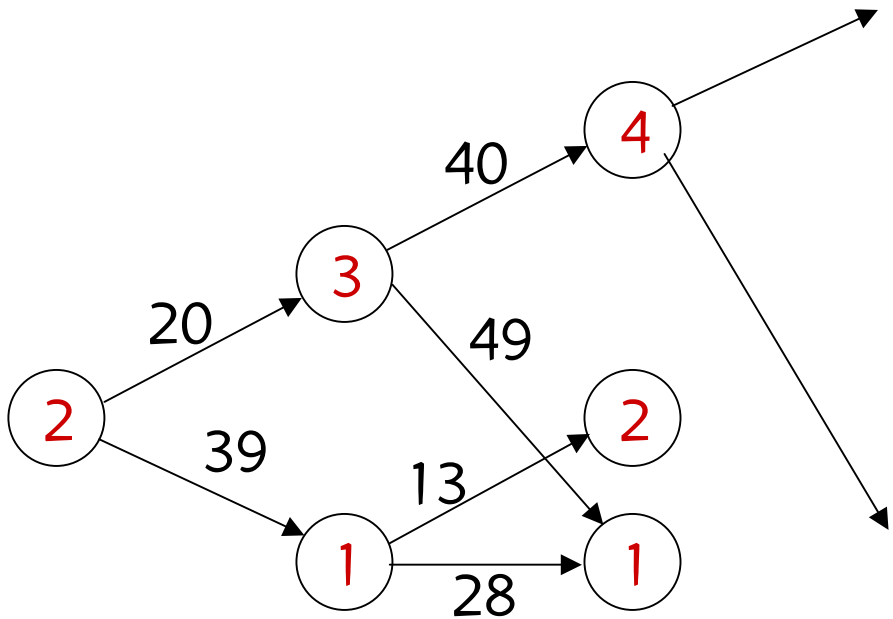
Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

Programação dinâmica



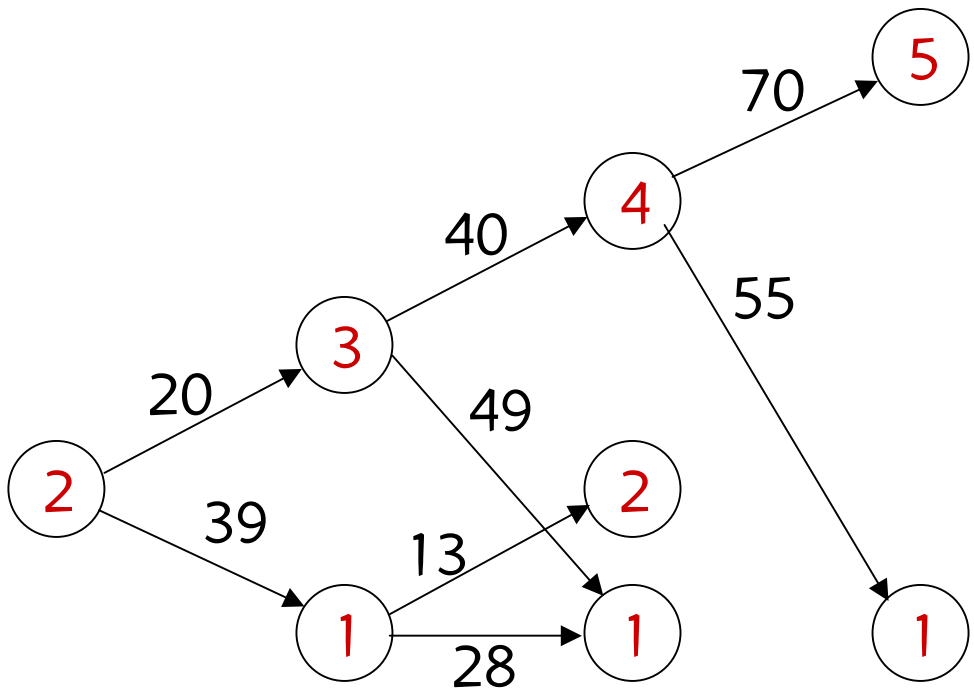
Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

Programação dinâmica



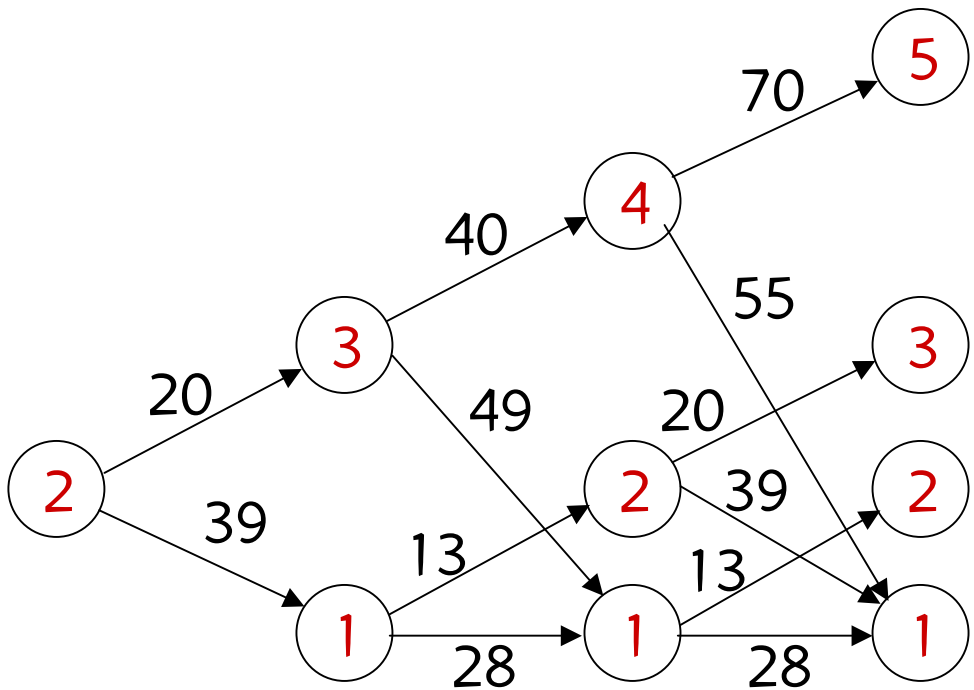
Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

Programação dinâmica



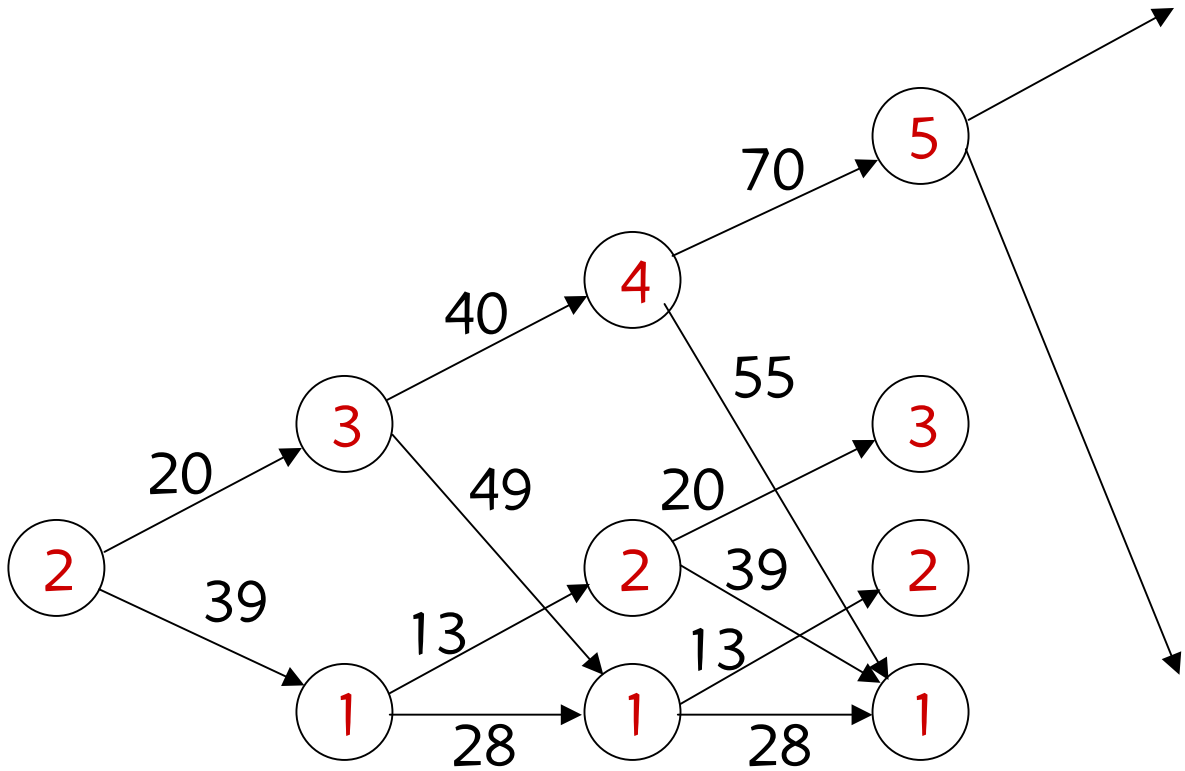
Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

Programação dinâmica

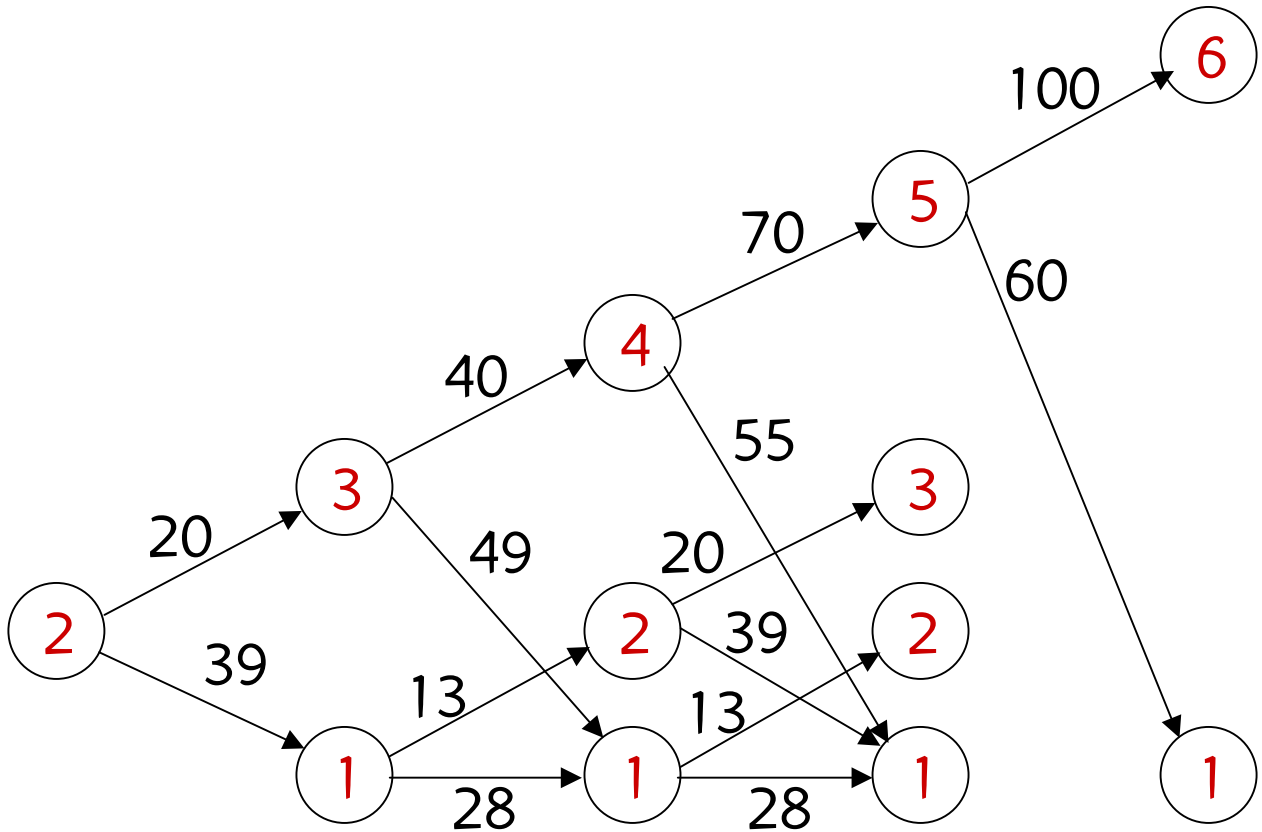


Ano 1 Ano 2 Ano 3 Ano 4 Ano N=5 Ano 6

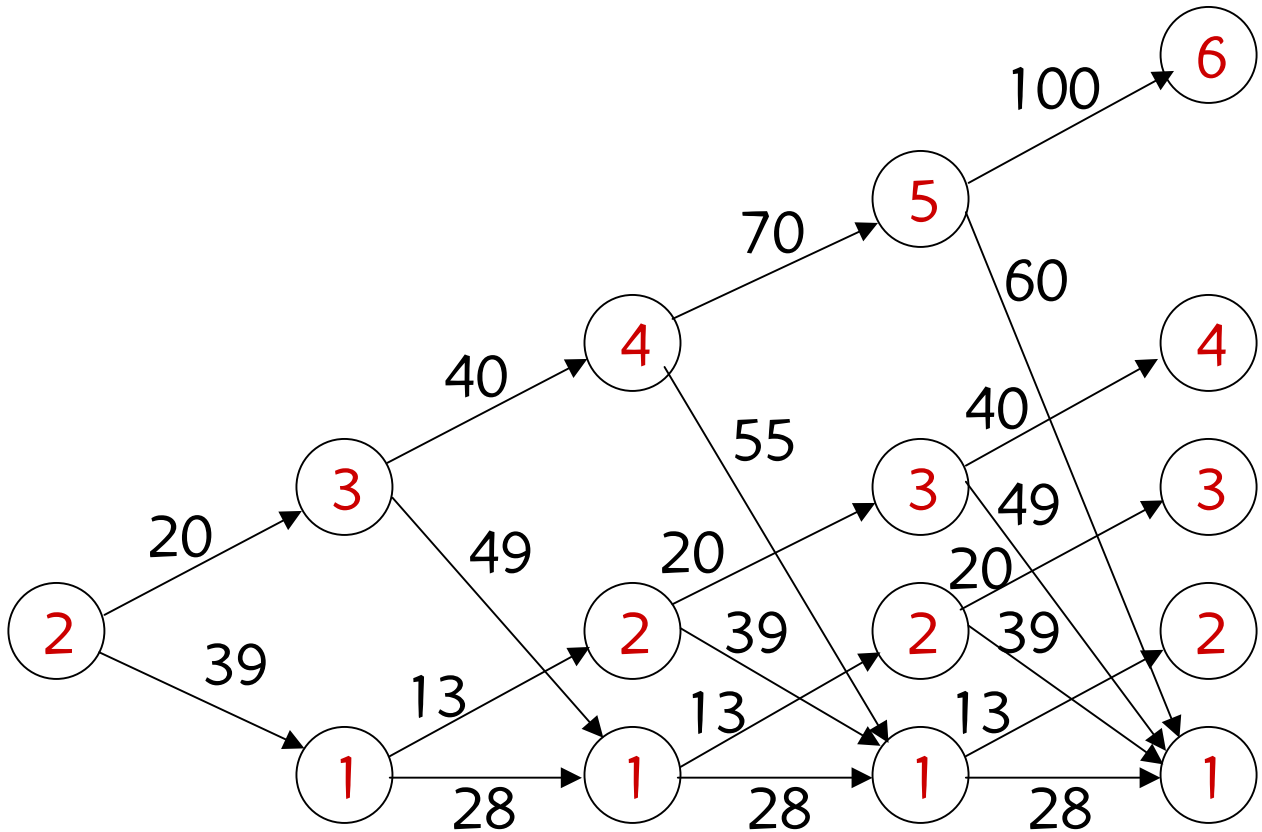
Programação dinâmica



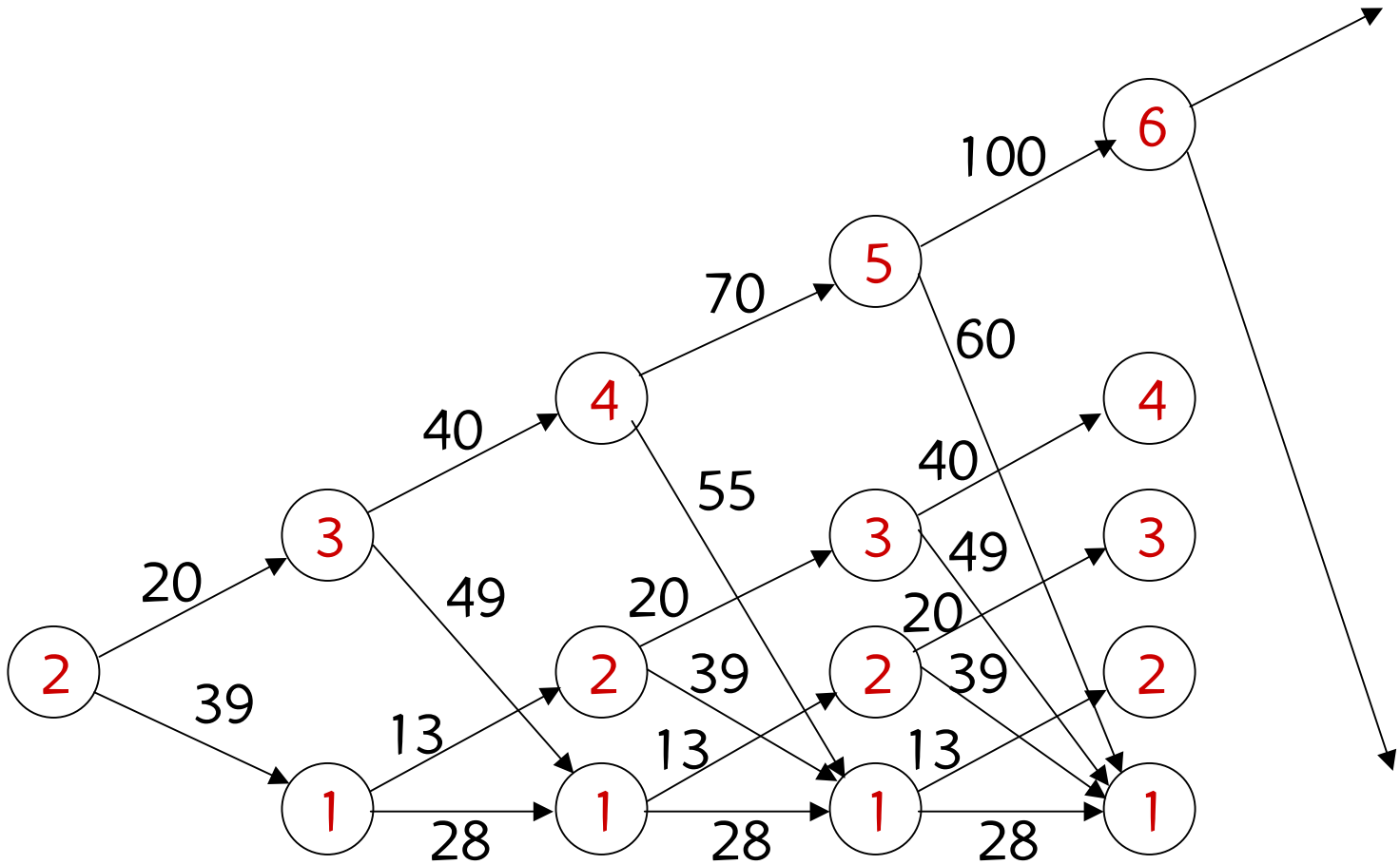
Programação dinâmica



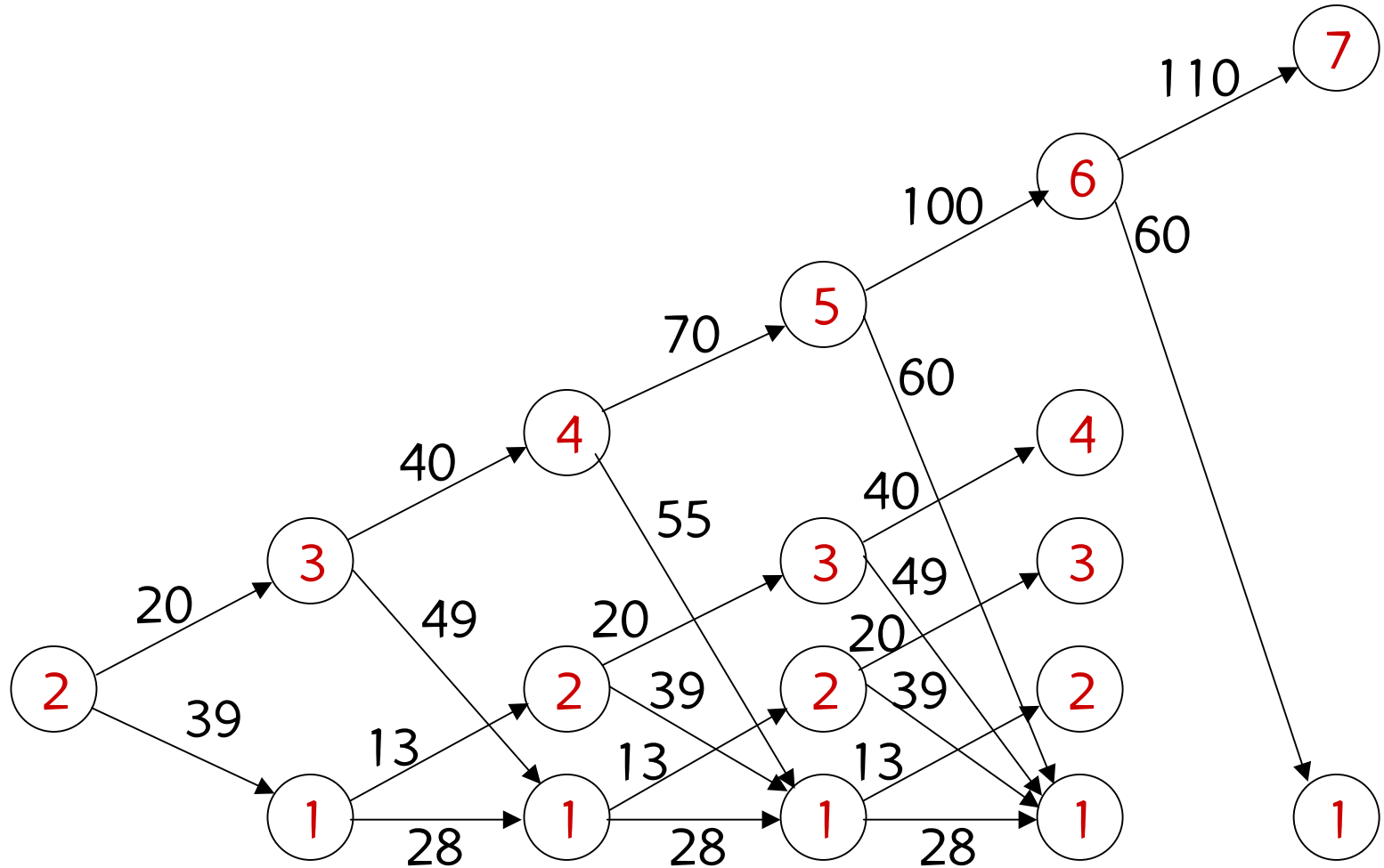
Programação dinâmica



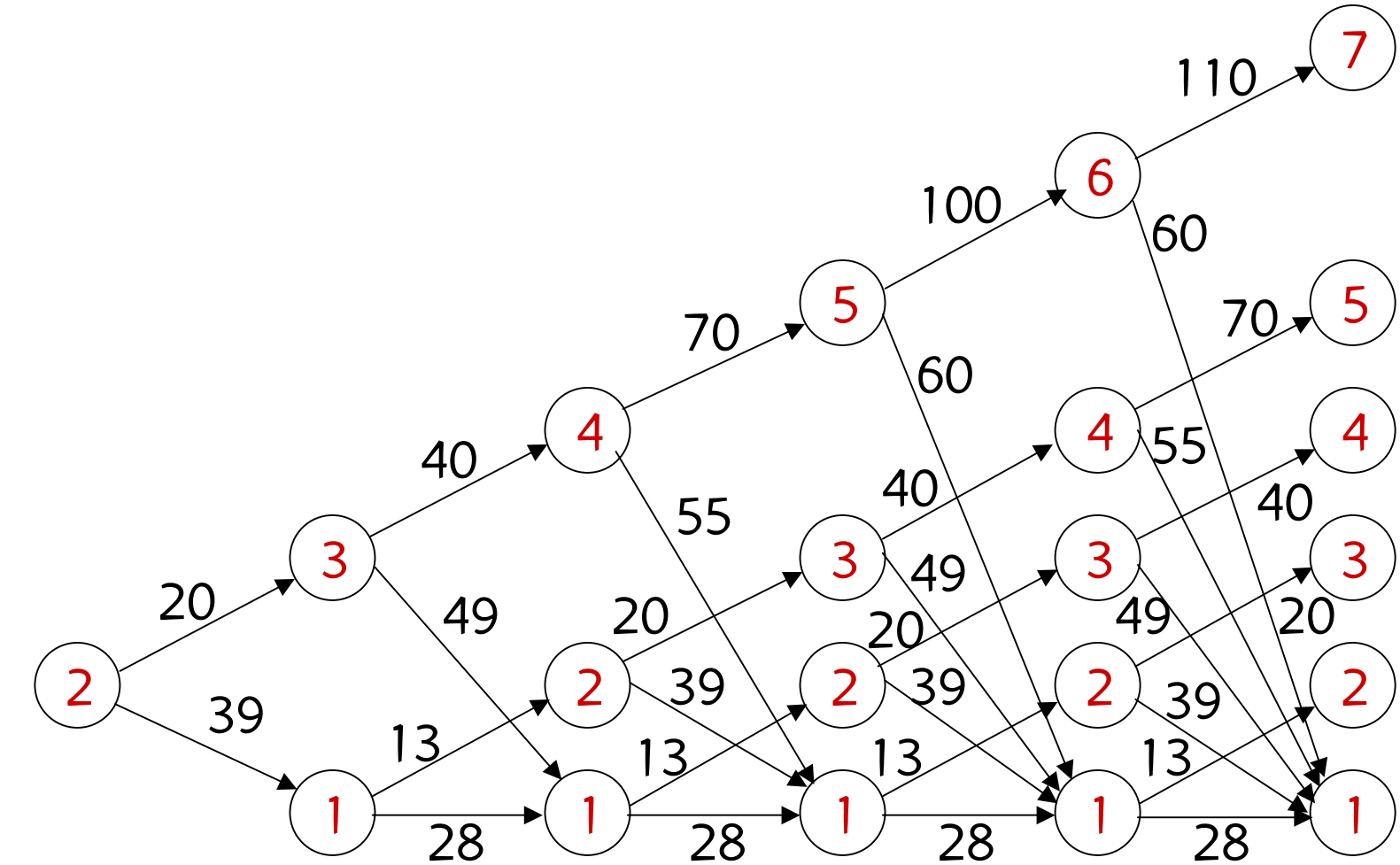
Programação dinâmica



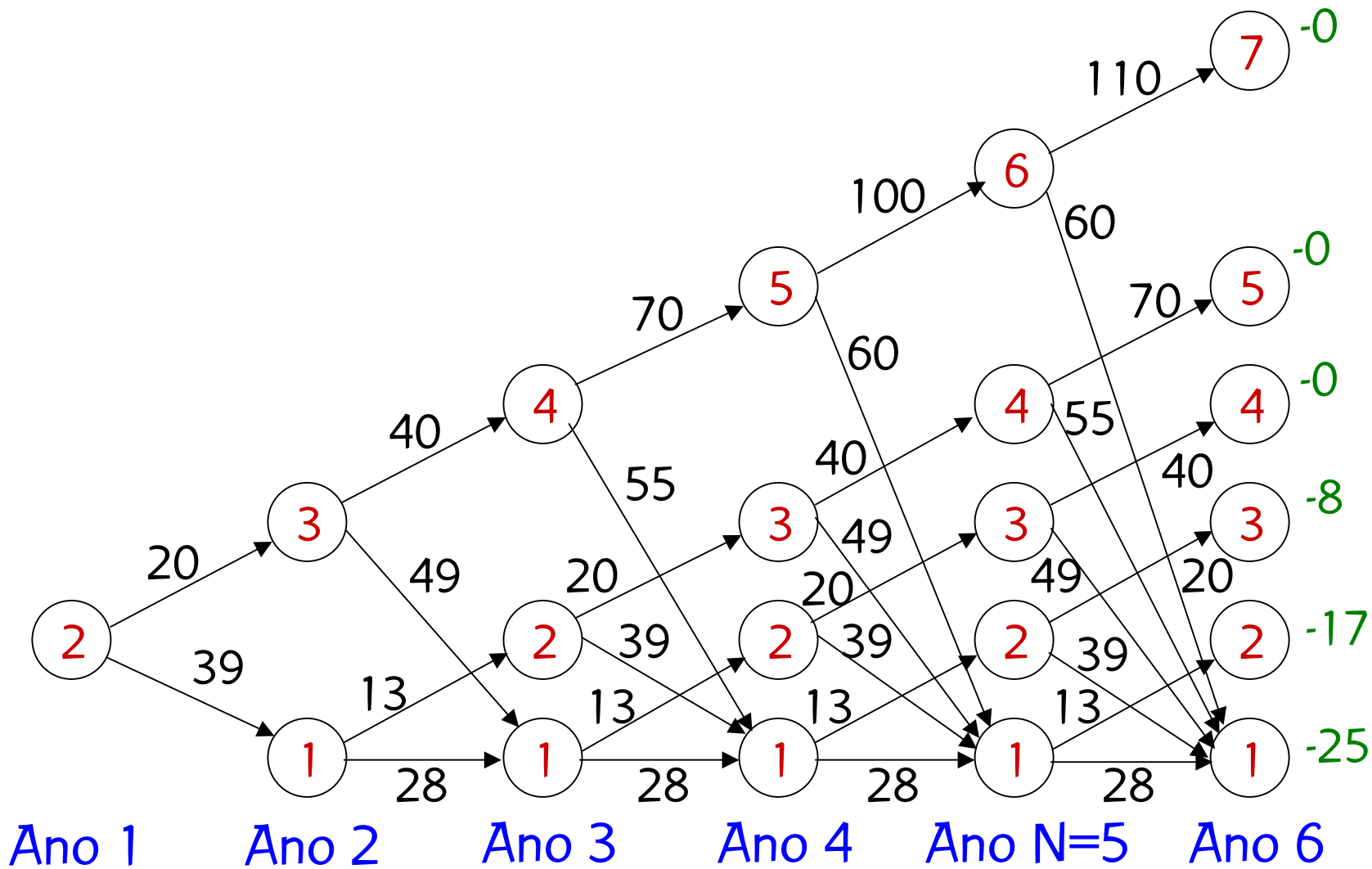
Programação dinâmica



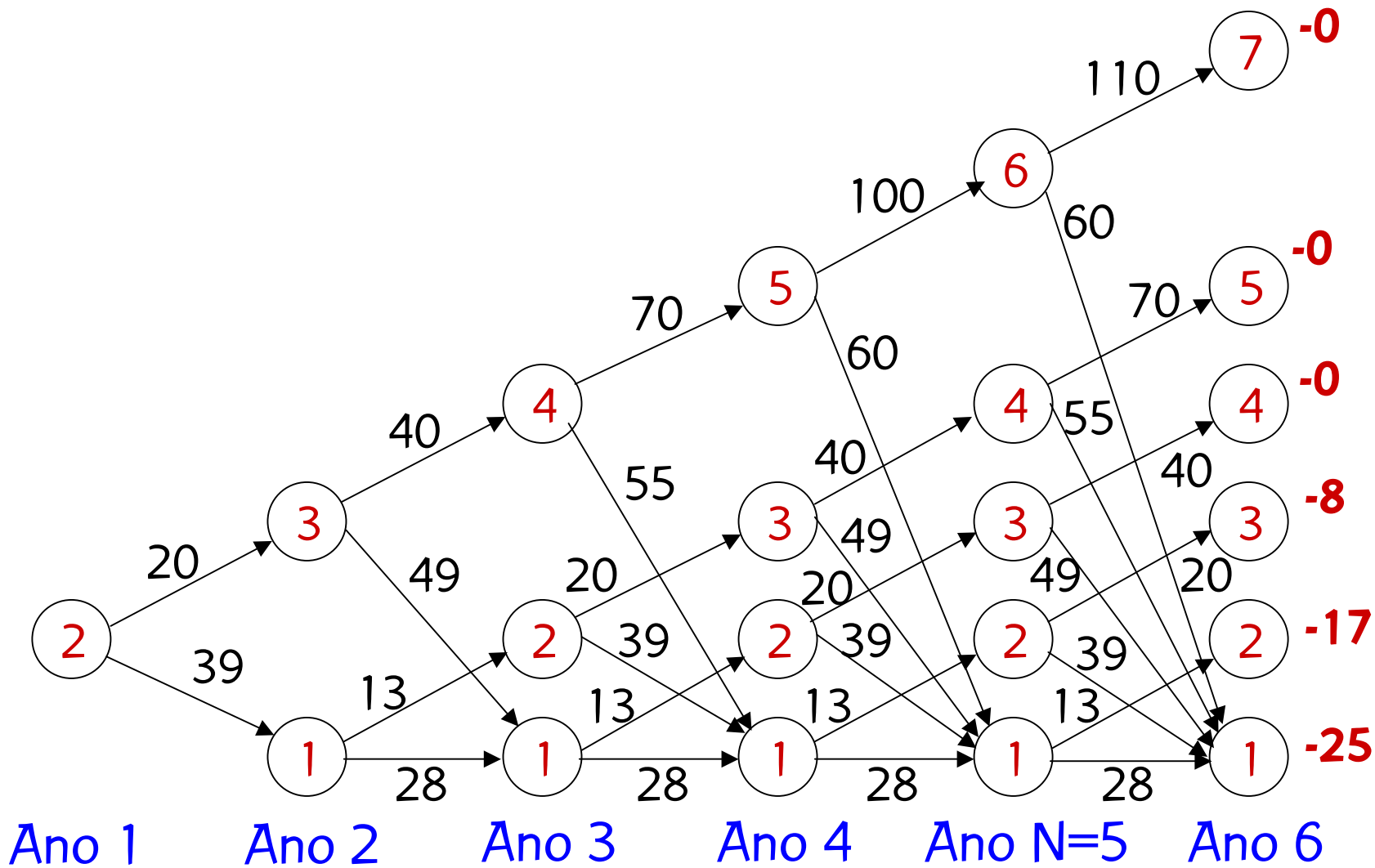
Programação dinâmica



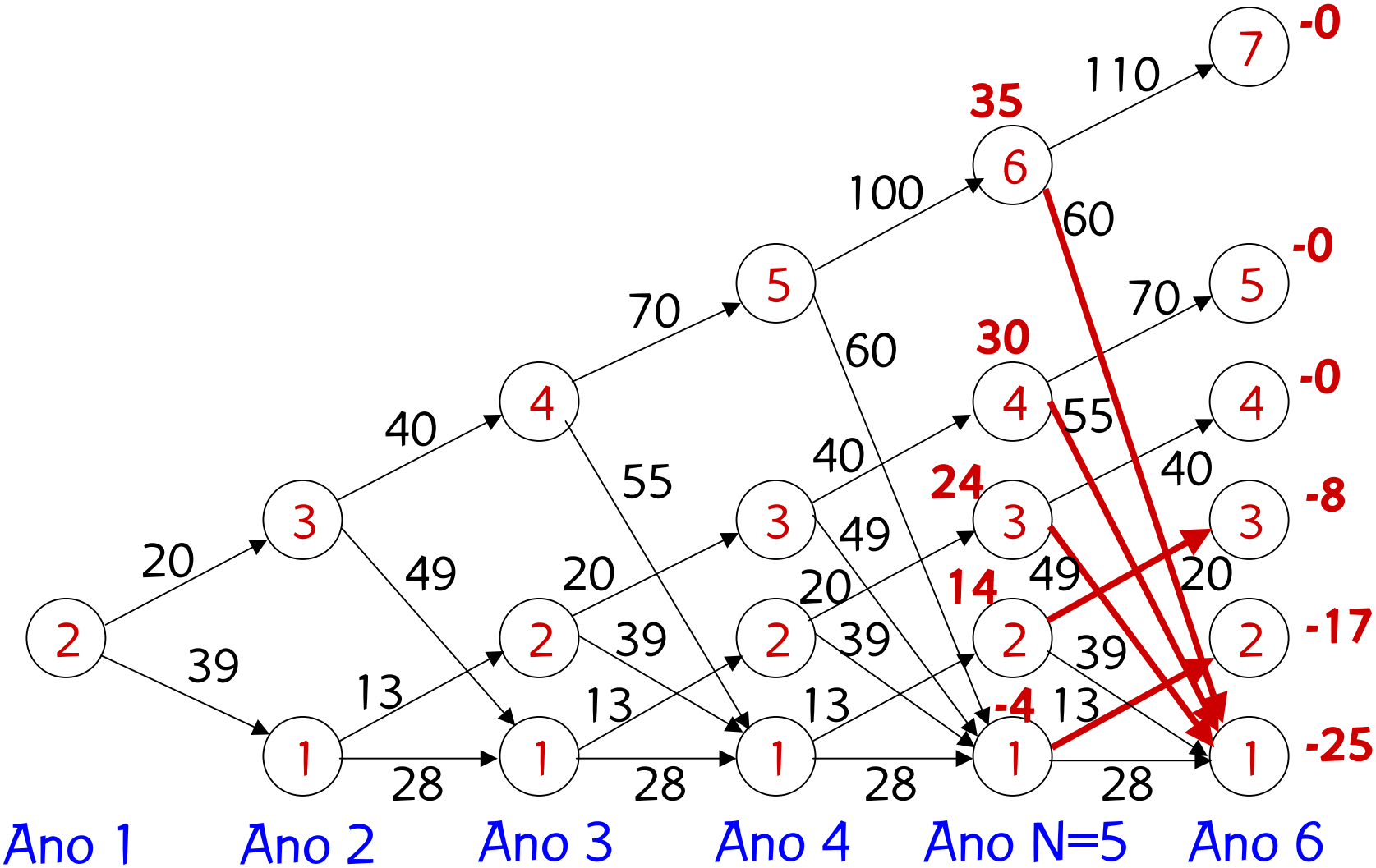
Programação dinâmica



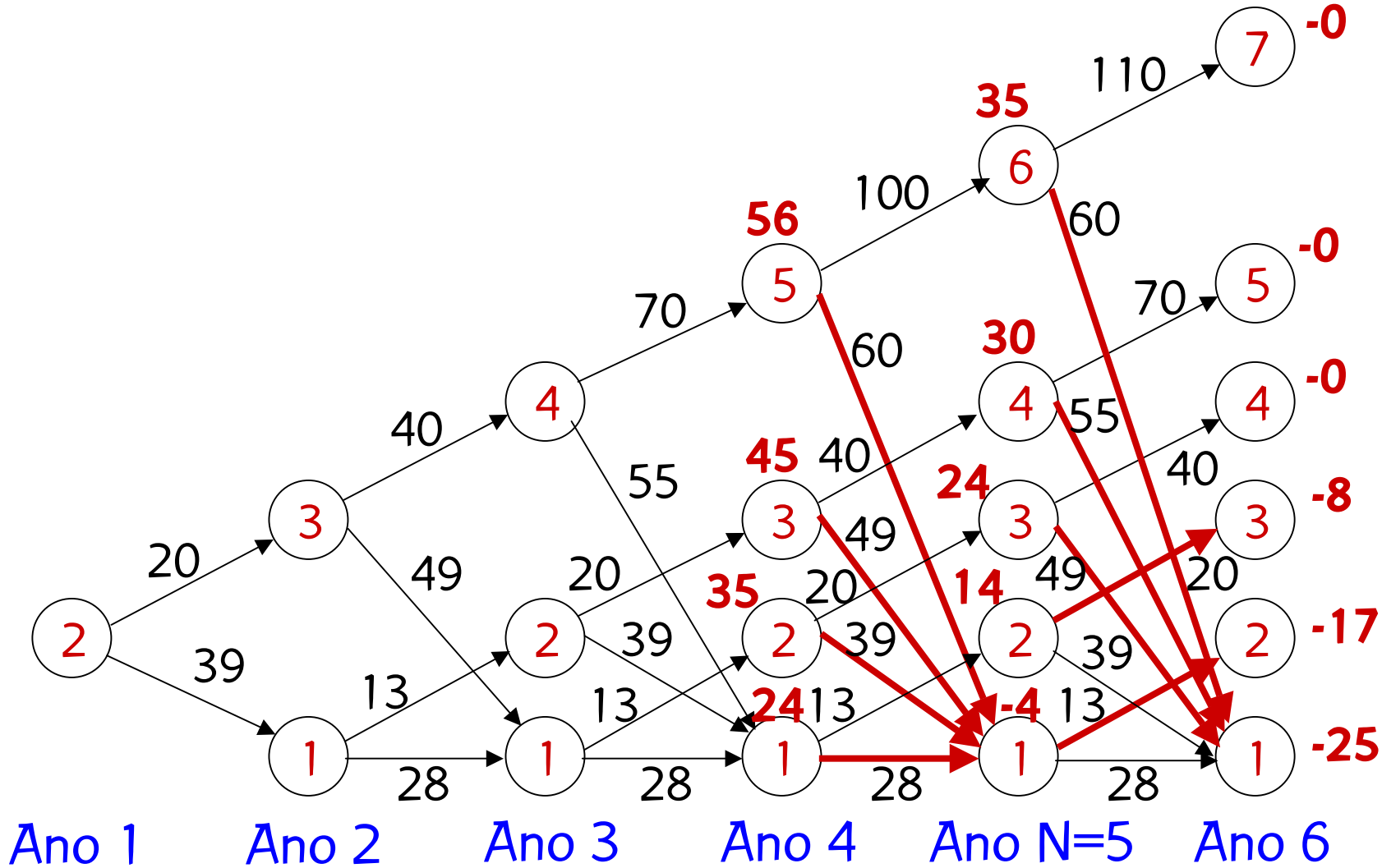
Programação dinâmica



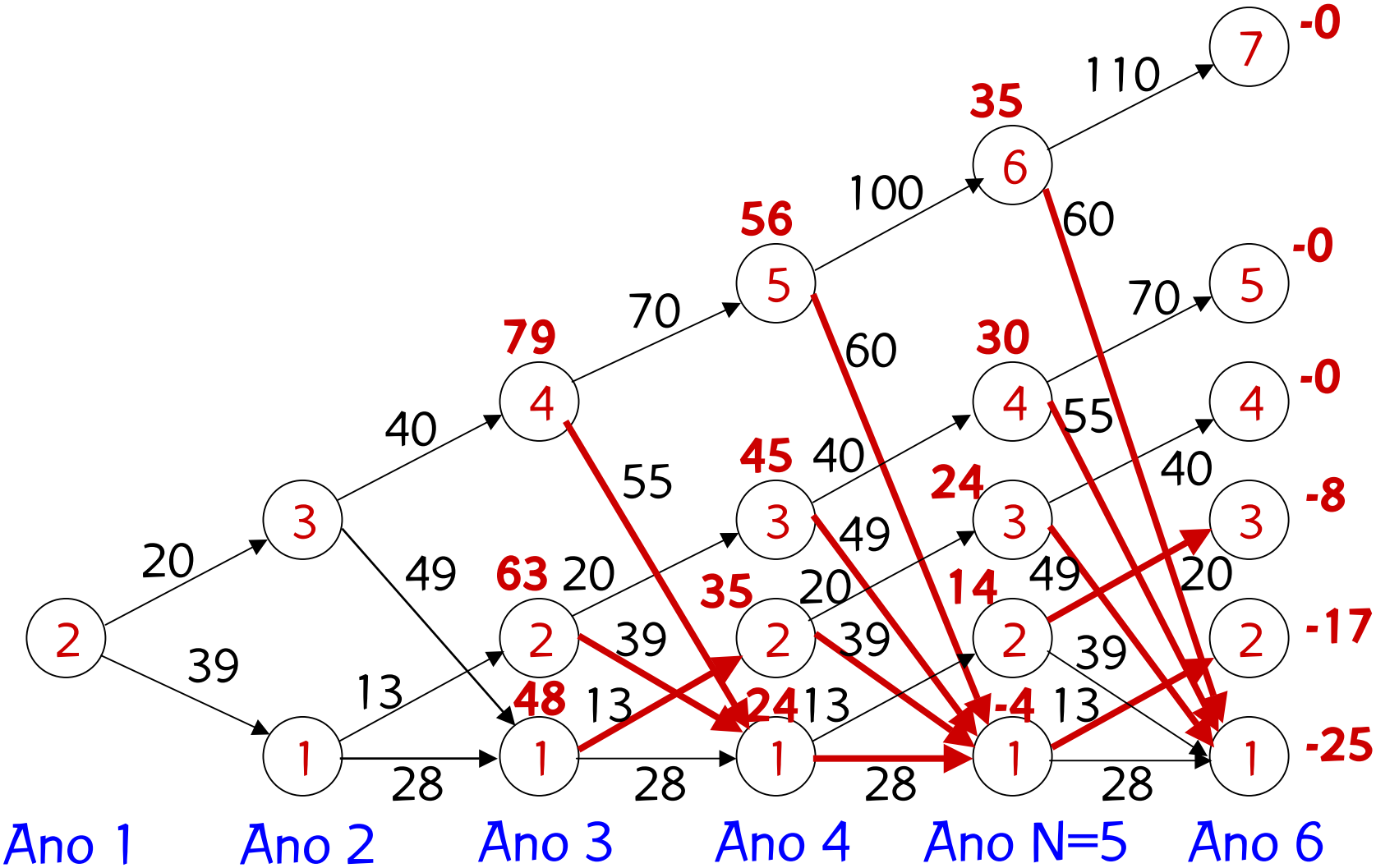
Programação dinâmica



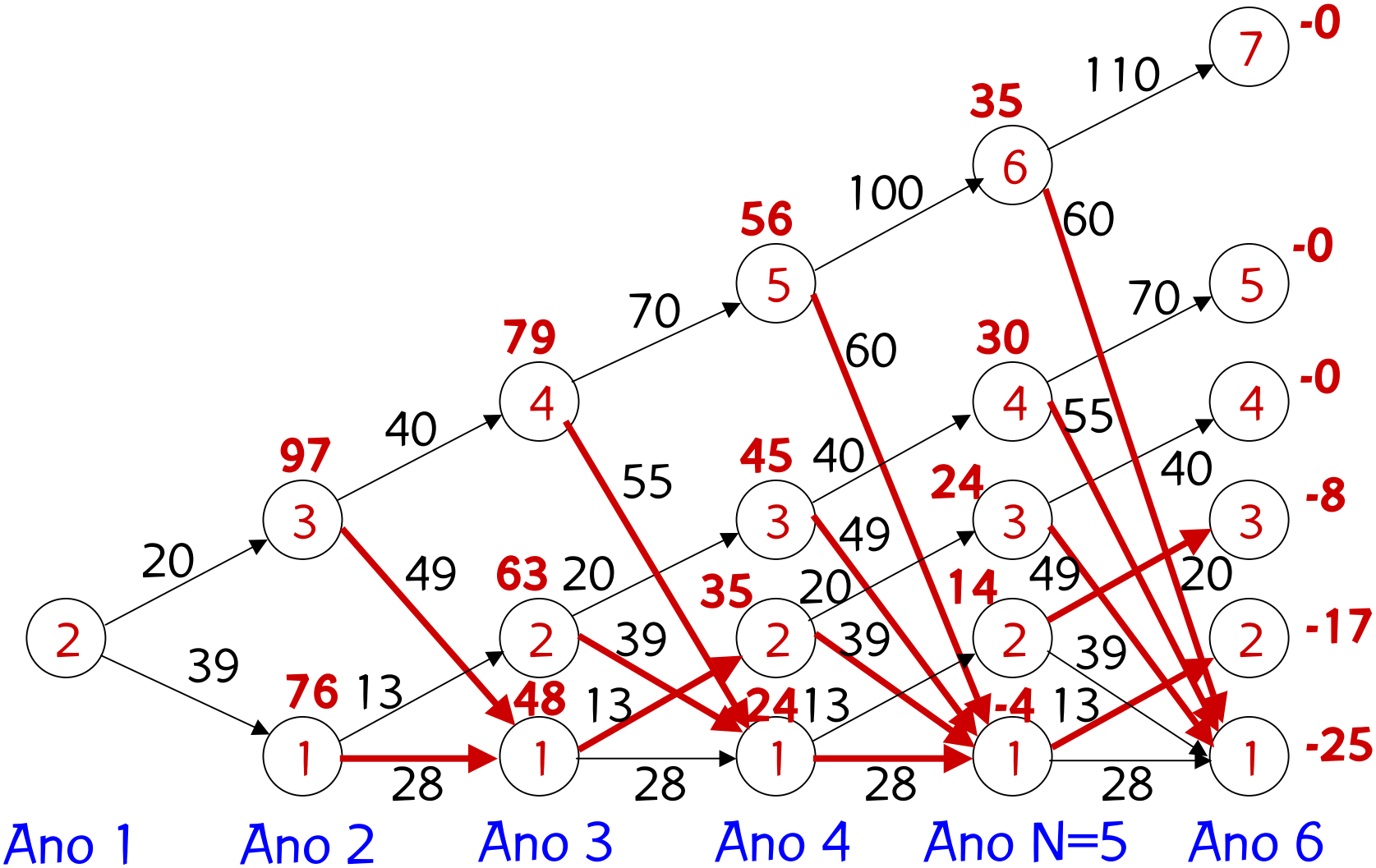
Programação dinâmica



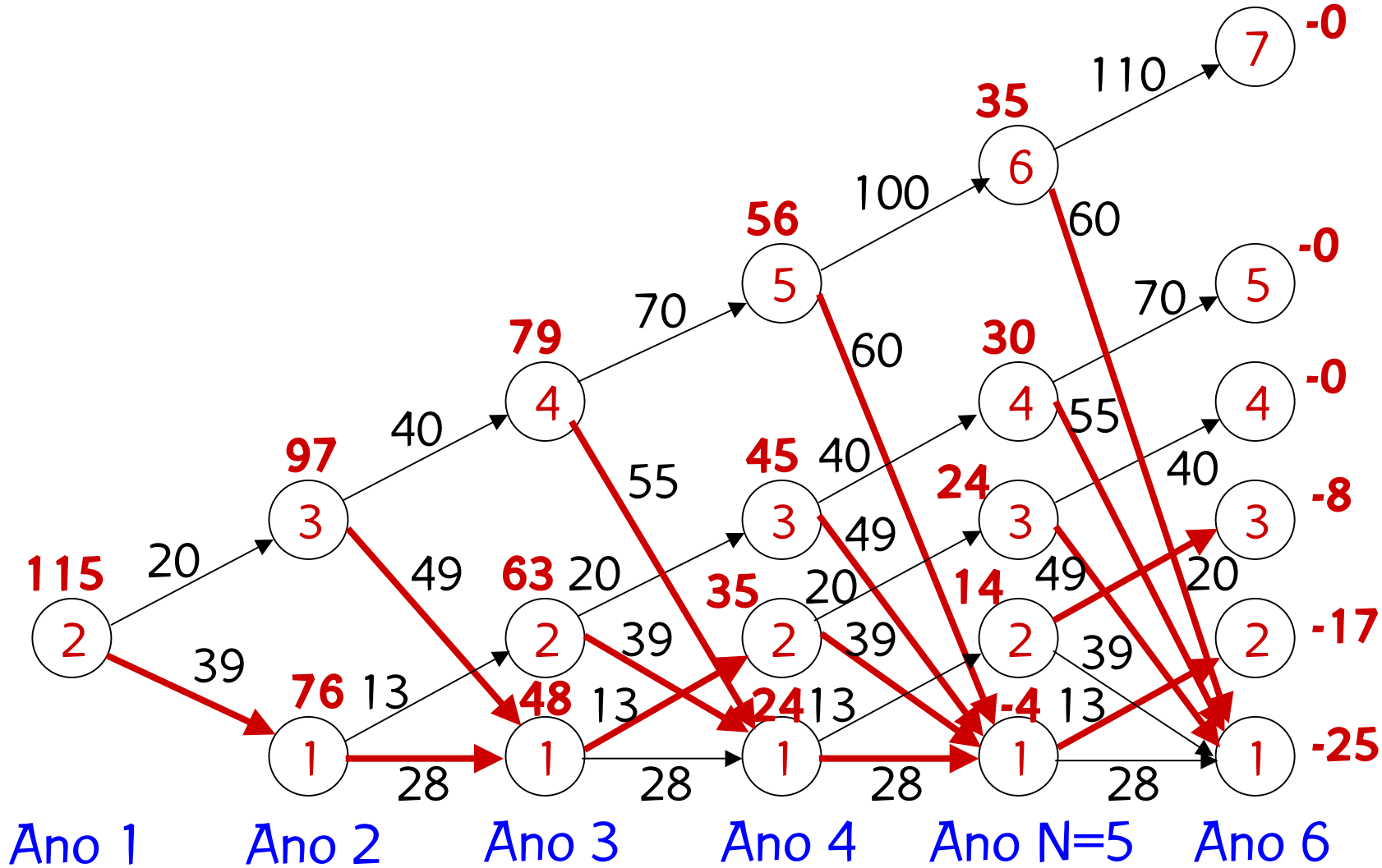
Programação dinâmica



Programação dinâmica

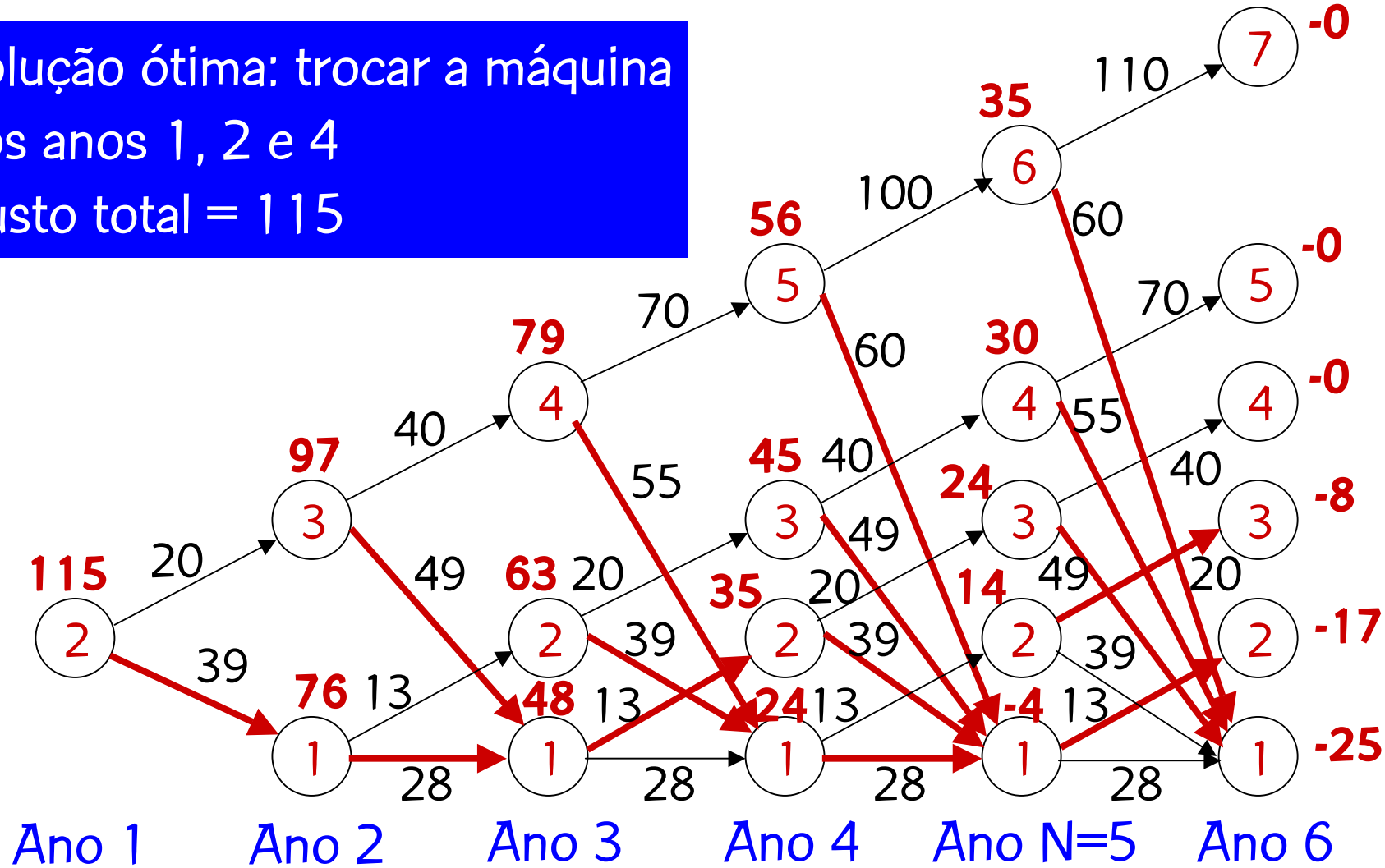


Programação dinâmica



Programação dinâmica

Solução ótima: trocar a máquina nos anos 1, 2 e 4
Custo total = 115



Programação dinâmica

- (III) Problema simplificado de planejamento da operação: Considere-se um sistema formado por uma usina hidráulica com reservatório (ou um sistema equivalente) e uma usina térmica capaz de complementar toda a demanda. São conhecidas as afluências mensais, as demandas mensais, a vazão turbinada máxima mensal da usina hidráulica e os volumes inicial e final do reservatório. Equacionar um modelo passível de ser resolvido por programação dinâmica para a otimização da operação, minimizando o custo de complementação térmica.

Programação dinâmica

- Dados:
  - Horizonte: k = 1, 2, ..., N
  - Demanda no mês k: d(k)
  - Afluência no mês k: a(k)
  - Vazão turbinada máxima no mês k: p(k)
  - Custo unitário da geração térmica no mês k: c(k)
  - Volume mínimo do reservatório no mês k: m(k)
  - Volume máximo do reservatório no mês k: M(k)
  - Volume inicial do reservatório: $V(1) = V_i$
  - Volume final do reservatório: $V(N+1) = V_f$

Programação dinâmica

- Estágio:
  - cada mês k = 1, 2, ..., N do horizonte de planejamento
- Estado:
  - volume x(k) do reservatório no mês k
- Decisão:
  - vazão turbinada u(k) na usina hidráulica no mês k

  geração térmica t(k) no mês k calculada a partir de u(k) e d(k):
  t(k) = max {0,d(k)-produção(u(k))}

Programação dinâmica

- Recursão:
  - $x(k+1) = x(k) + a(k) - u(k)$
  - $m(k) \leq x(k) \leq M(k)$
  - $x(1) = V_i$
  - $x(N+1) = V_f$
  - $0 \leq u(k) \leq p(k)$
  - $t(k) = \max \{0, d(k) - \text{produção}(u(k))\}$
  - minimizar $\Sigma_{k=1,\ldots,n}\ c(k).t(k)$

Programação dinâmica

- Considerações:
  - Não-linearidades
    - produção na usina hidráulica em função da vazão turbinada
    - custo de complementação térmica
  - Discretização dos volumes e das vazões turbinadas
    - aumento do espaço de estados
  - Outras variáveis de decisão
    - vertimento v(k) no mês k: x(k+1) = x(k) + a(k) - u(k) - v(k)
    - produção térmica no mês k: d(k) = produção(u(k)) + t(k) + deficit(k)
    - aumento do espaço de decisões

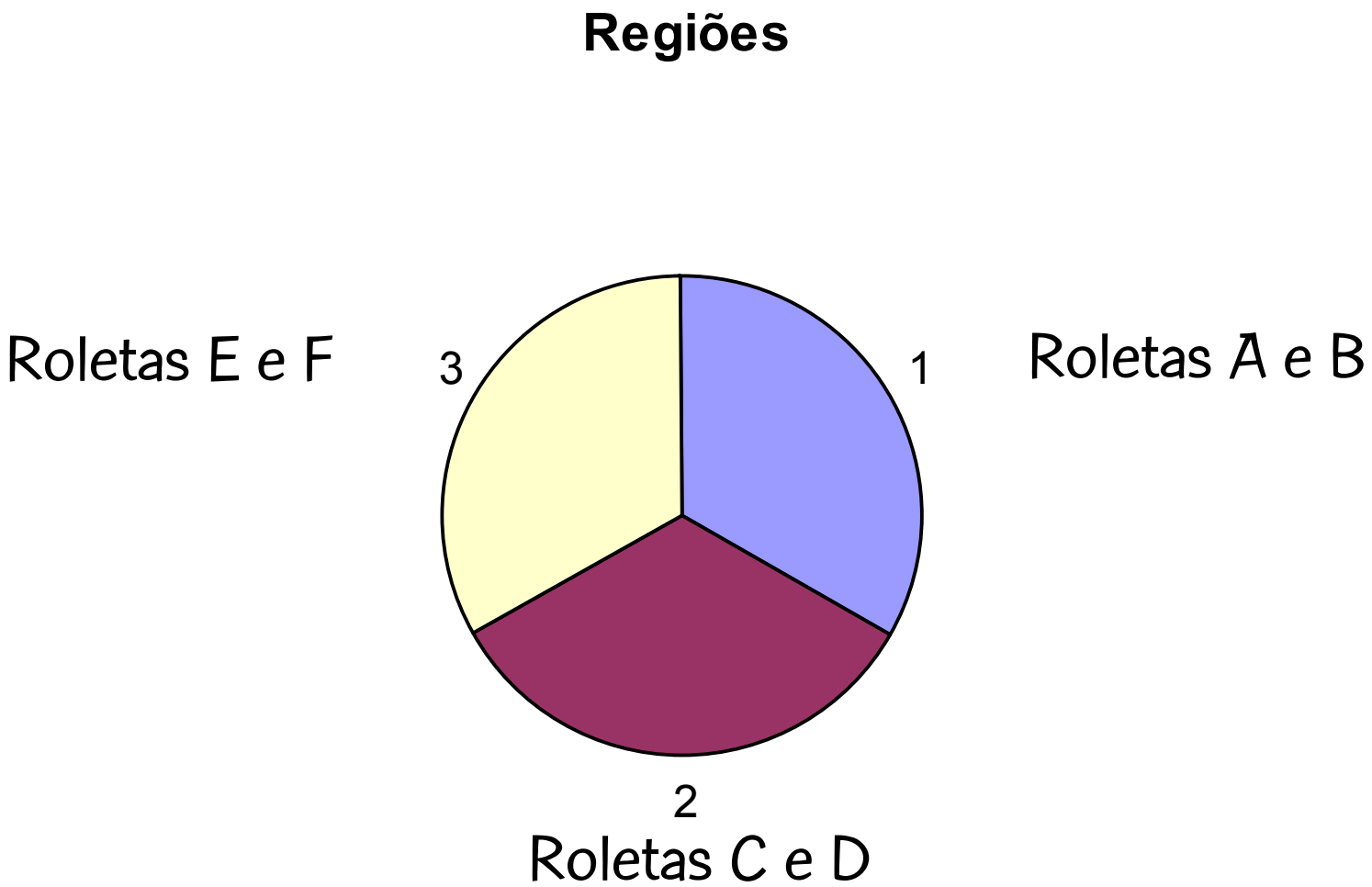
Programação dinâmica

- Considerações (continuação):
  - Múltiplos reservatórios
    - estado do sistema descrito por um vetor de volumes
    - aumento do espaço de estados
    - "explosão combinatória"
  - Demandas e/ou afluências não-determinísticas
    - modelo de programação dinâmica estocástica

Programação dinâmica estocástica

- Motivação: Considera-se o seguinte jogo de roletas. Um tabuleiro está dividido em três regiões iguais e equiprováveis. O jogador inicia o jogo comprando uma ficha e lançando-a ao acaso sobre o tabuleiro. Em cada região há duas roletas. O jogador deve escolher e girar uma das roletas associadas à região onde caiu a ficha. O resultado da roleta indica a nova região onde deve ser colocada a ficha. São conhecidas a probabilidade e o ganho de cada transição de região para cada roleta.

Programação dinâmica estocástica



Programação dinâmica estocástica

O jogador deve repetir o procedimento anterior por mais duas jogadas, escolhendo sempre uma das roletas associadas à região onde caiu a ficha. Determinar quais devem ser as roletas escolhidas, de modo a maximizar o valor esperado de seus ganhos após as três jogadas.

Programação dinâmica estocástica

- Região 1:

Roleta	Transição	Probabilidade	Ganho
A	1	1/4	12
	2	3/4	4
B	1	1/6	0
	3	5/6	6

Programação dinâmica estocástica

- Região 2:

Roleta	Transição	Probabilidade	Ganho
C	2	1/2	10
	1	1/2	10
D	2	1/4	0
	3	3/4	12

Programação dinâmica estocástica

- Região 3:

Roleta	Transição	Probabilidade	Ganho
E	3	3/4	8
	1	1/4	36
F	3	1/2	16
	2	1/2	10

Programação dinâmica estocástica

- Estágios:
  - cada uma das jogadas k = 1, 2, 3
- Estados:
  - região onde a ficha se encontra
- Decisões:
  - escolha de uma das duas roletas em cada região
  - após a escolha de uma das roletas, ocorre uma fenômeno aleatório: transição de uma roleta para outra

Programação dinâmica estocástica

- É como se cada decisão tivesse duas componentes:
  - escolha determinística da roleta
  - transição aleatória para outra região
- Conseqüência: não é possível calcular o ganho associado a cada decisão
- Mas é possível calcular o ganho esperado associado a cada decisão!

Programação dinâmica estocástica

- Quando falta uma única jogada, não é possível prever em que região a ficha se encontrará.
- No problema de PD determinística, o estado não é conhecido pois a política ótima não é conhecida a priori.
- No problema de PD probabilística, o estado não é conhecido por mais uma razão adicional: não se conhece o resultado (transição) de cada decisão.
- Entretanto, pode-se calcular o ganho adicional esperado para cada roleta escolhida.

Programação dinâmica estocástica

- Caso a ficha encontre-se na região 1 na última jogada:

  Ganho esperado com a escolha da roleta A:

  (1/4) x 12 + (3/4) x 4 = 6

  Ganho esperado com a escolha da roleta B:

  (1/6) x 0 + (5/6) x 6 = 5

  A escolha da roleta A proporciona um maior ganho esperado.

Programação dinâmica estocástica

- Caso a ficha encontre-se na região 2 na última jogada:

  Ganho esperado com a escolha da roleta C:

  (1/2) x 10 + (1/2) x 10 = 10

  Ganho esperado com a escolha da roleta D:

  (1/4) x 0 + (3/4) x 12 = 9

  A escolha da roleta C proporciona um maior ganho esperado.

Programação dinâmica estocástica

- Caso a ficha encontre-se na região 3 na última jogada:

  Ganho esperado com a escolha da roleta E:

  (3/4) x 8 + (1/4) x 36 = 15

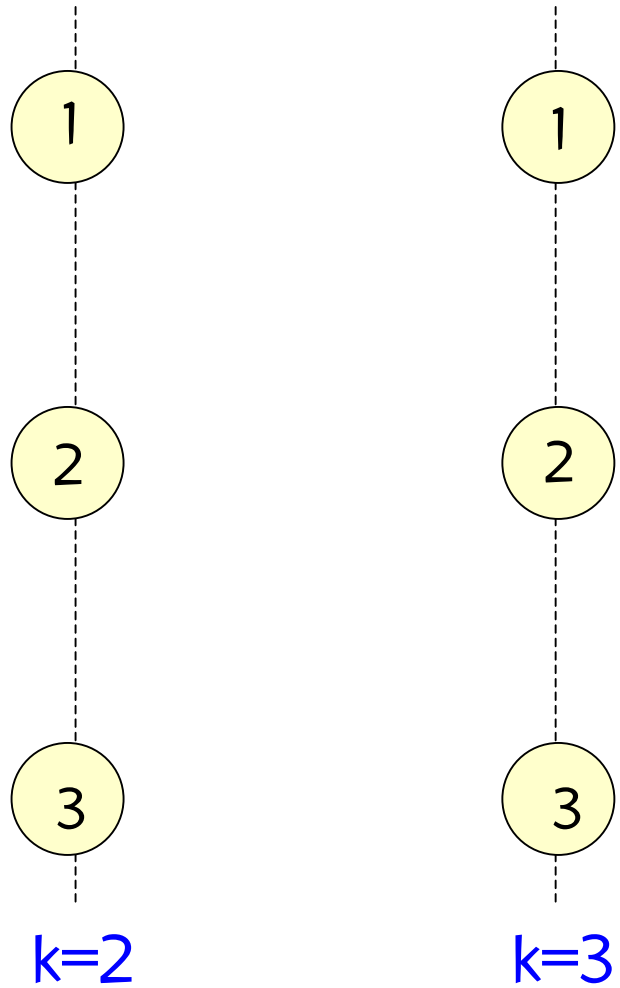
  Ganho esperado com a escolha da roleta F:

  (1/2) x 16 + (1/2) x 10 = 13

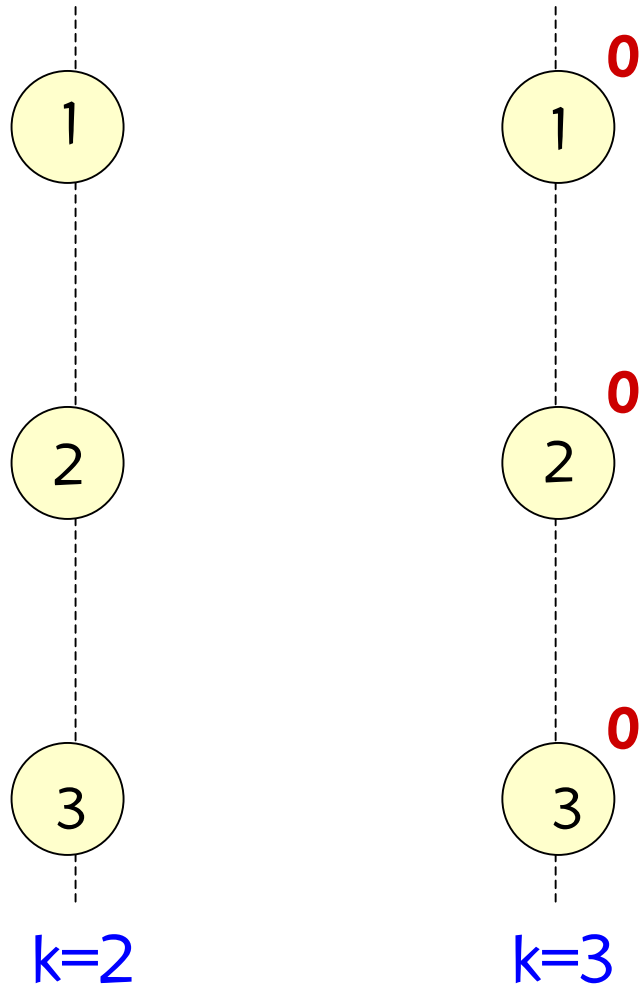
  A escolha da roleta E proporciona um maior ganho esperado.

- Se o jogador só tivesse direito a uma jogada, estas seriam suas decisões ótimas.

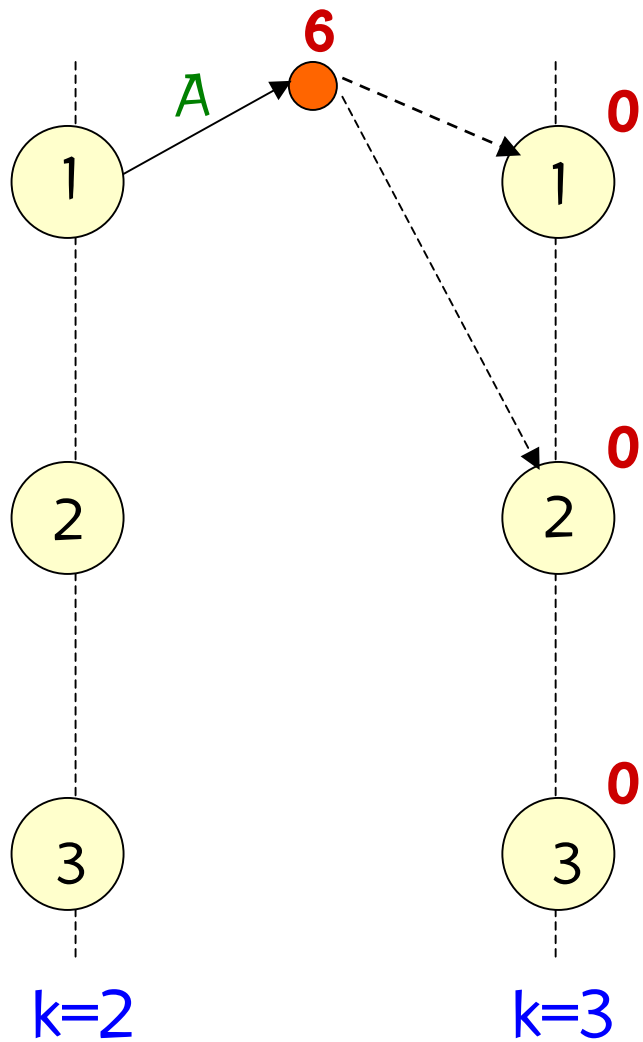
Programação dinâmica estocástica



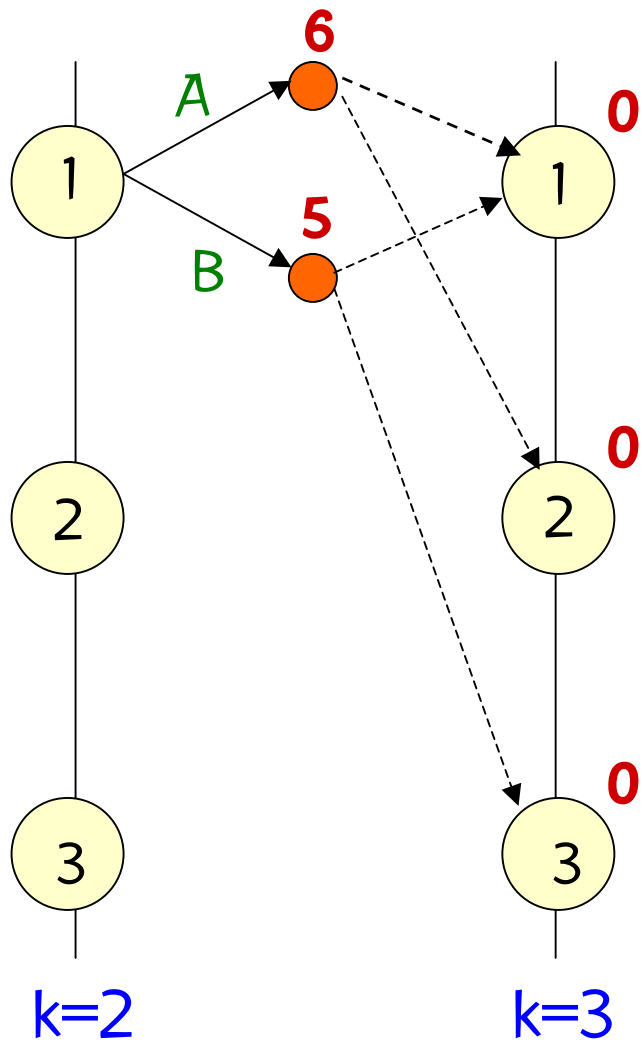
Programação dinâmica estocástica



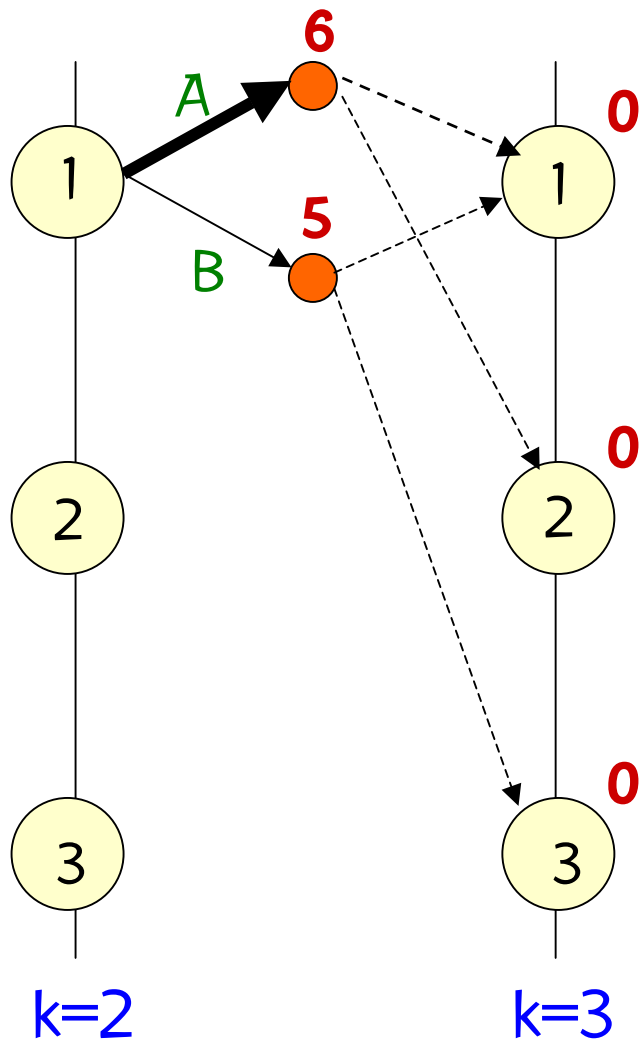
Programação dinâmica estocástica



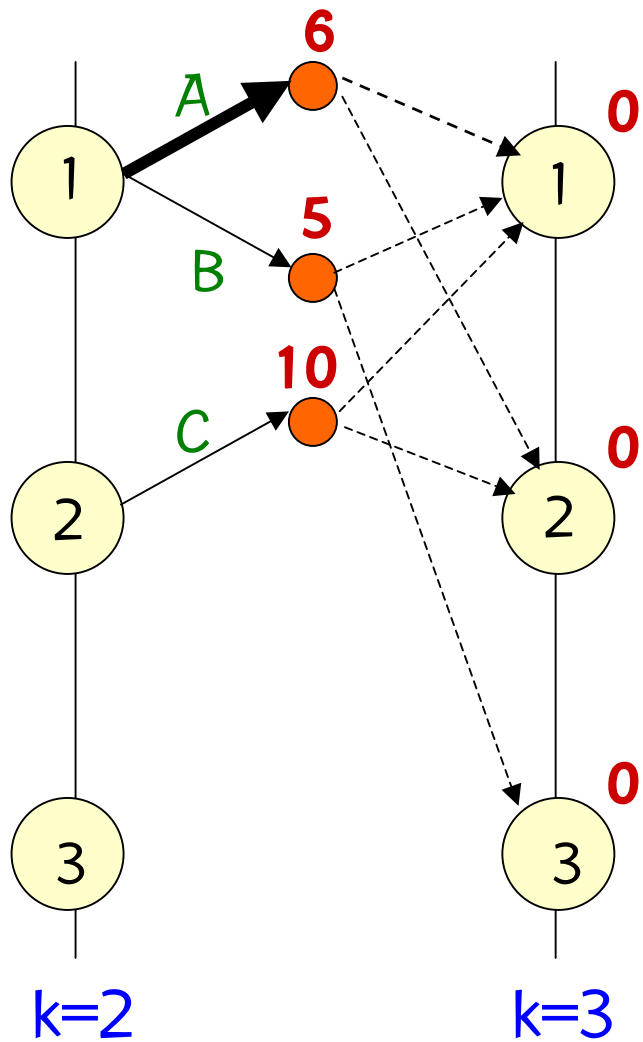
Programação dinâmica estocástica



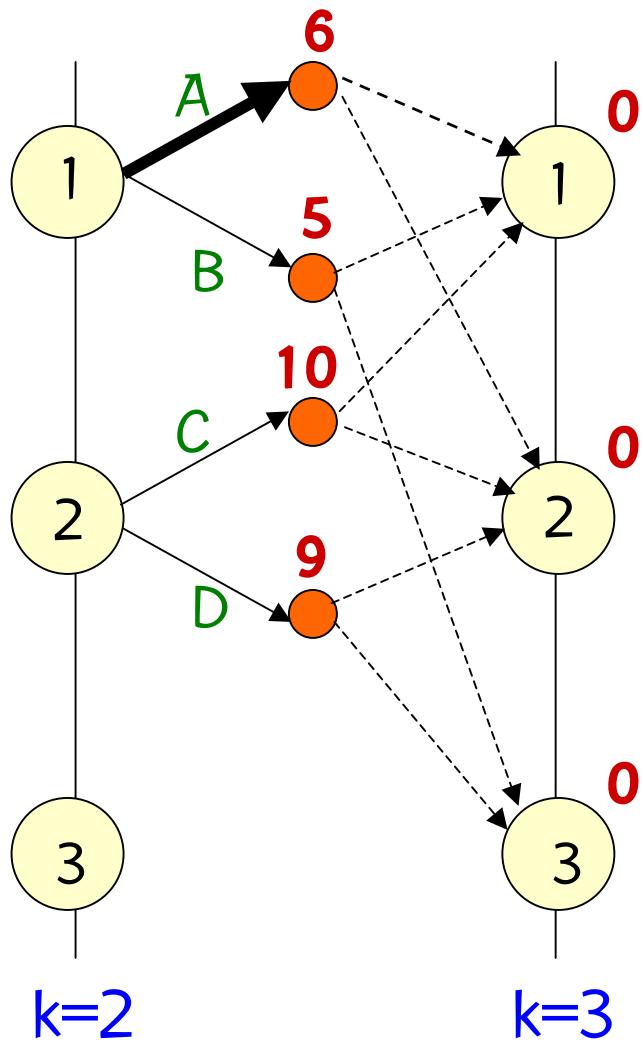
Programação dinâmica estocástica



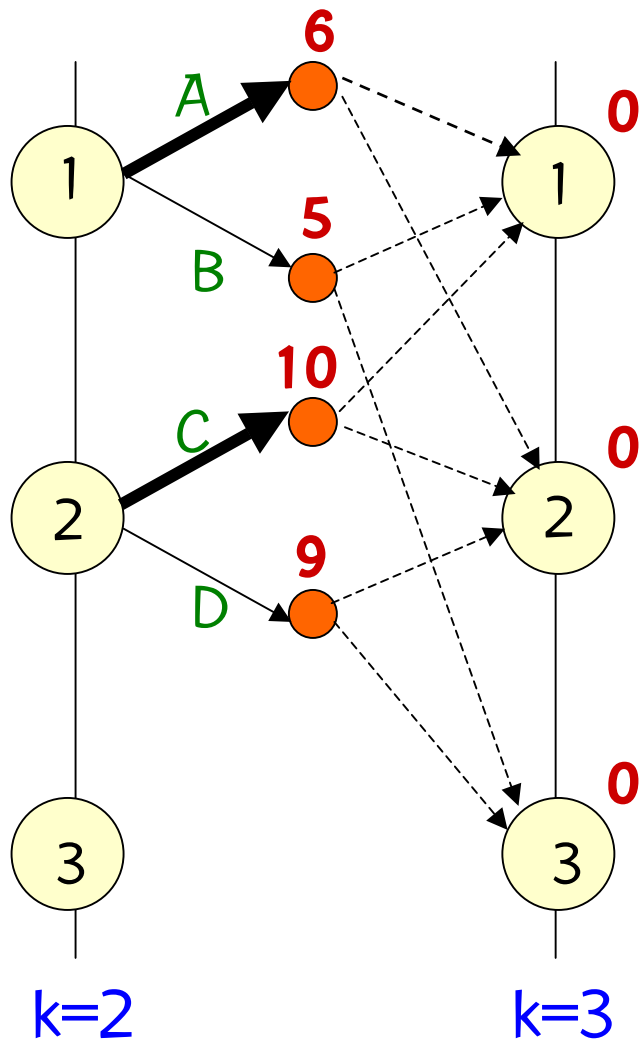
Programação dinâmica estocástica



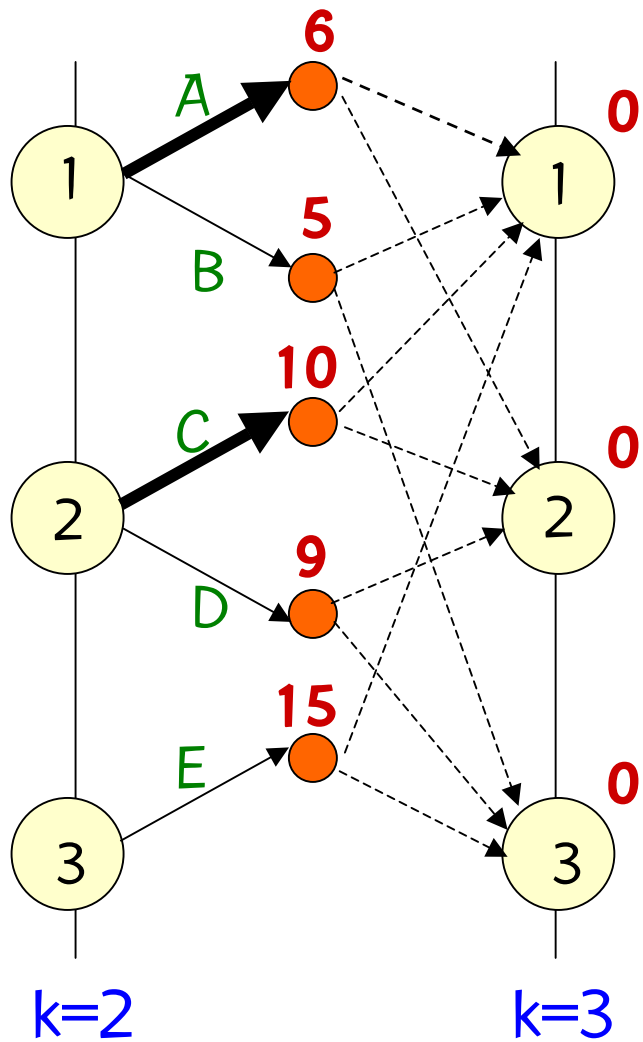
Programação dinâmica estocástica



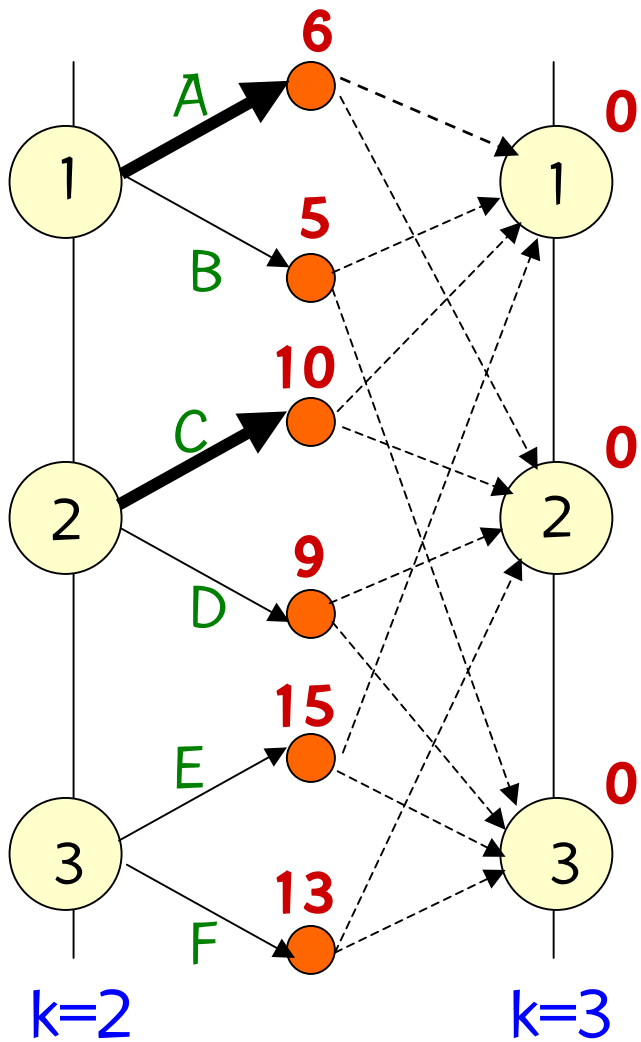
Programação dinâmica estocástica



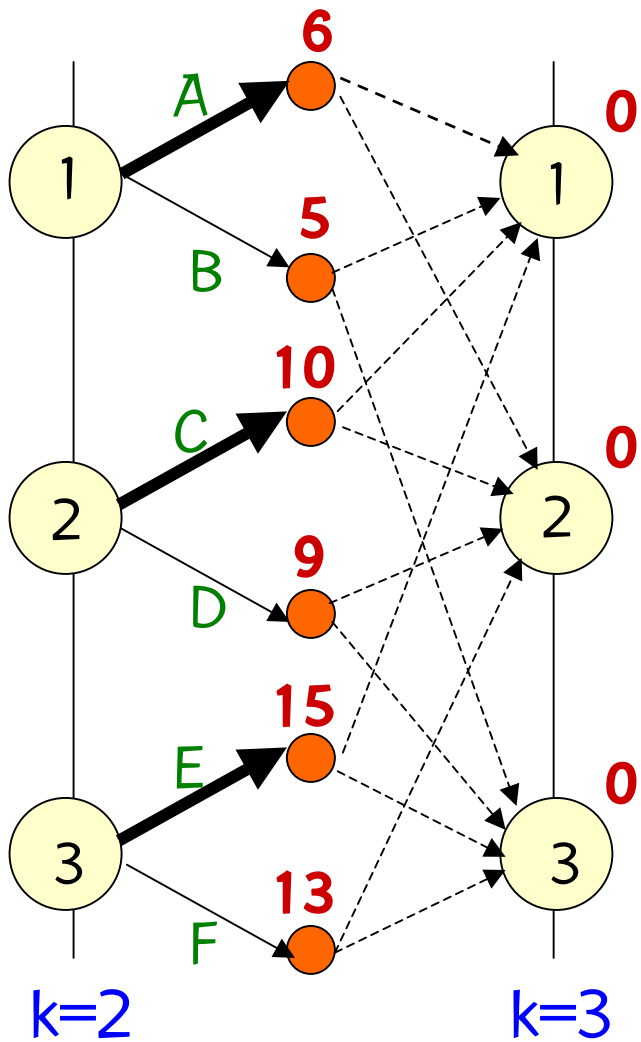
Programação dinâmica estocástica



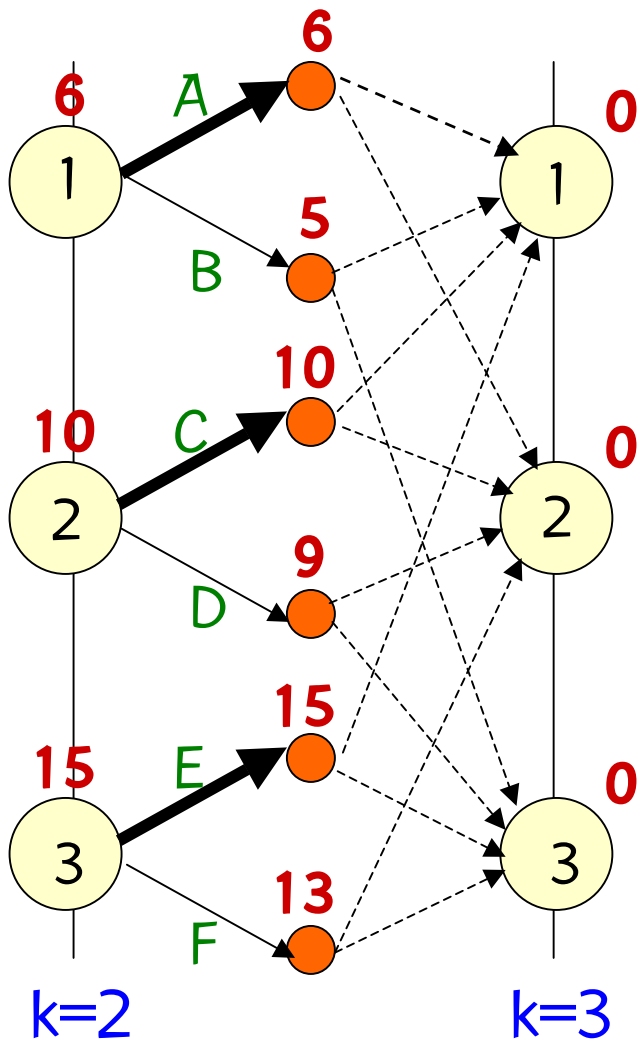
Programação dinâmica estocástica



Programação dinâmica estocástica



Programação dinâmica estocástica



Programação dinâmica estocástica

- Caso a ficha encontre-se na região 1 na segunda jogada:

  Ganho esperado com a escolha da roleta A:

  (1/4) x (12+6) + (3/4) x (4+10) = 15

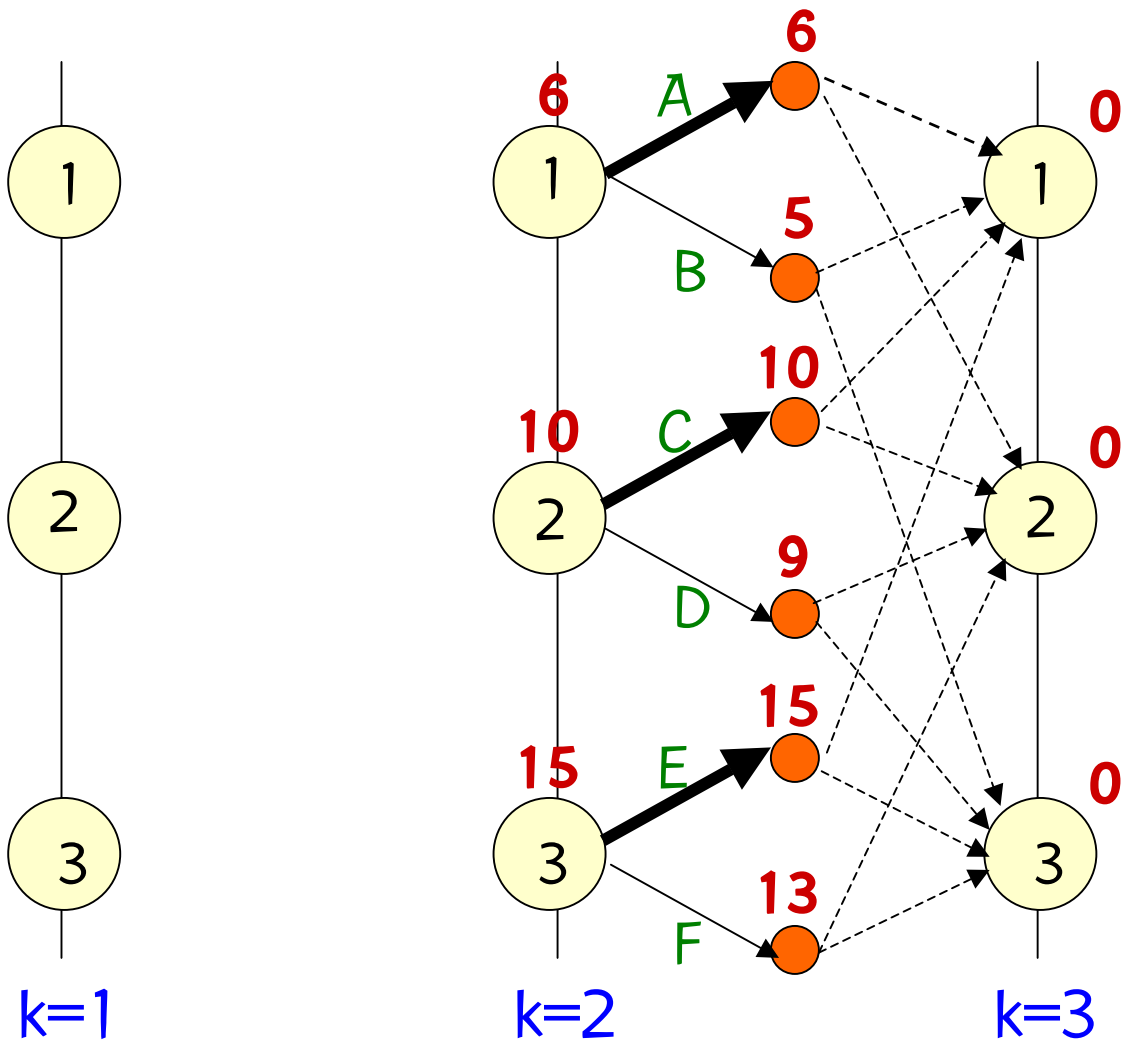
  Ganho esperado com a escolha da roleta B:

  (1/6) x (0+6) + (5/6) x (6+15) = 18,5

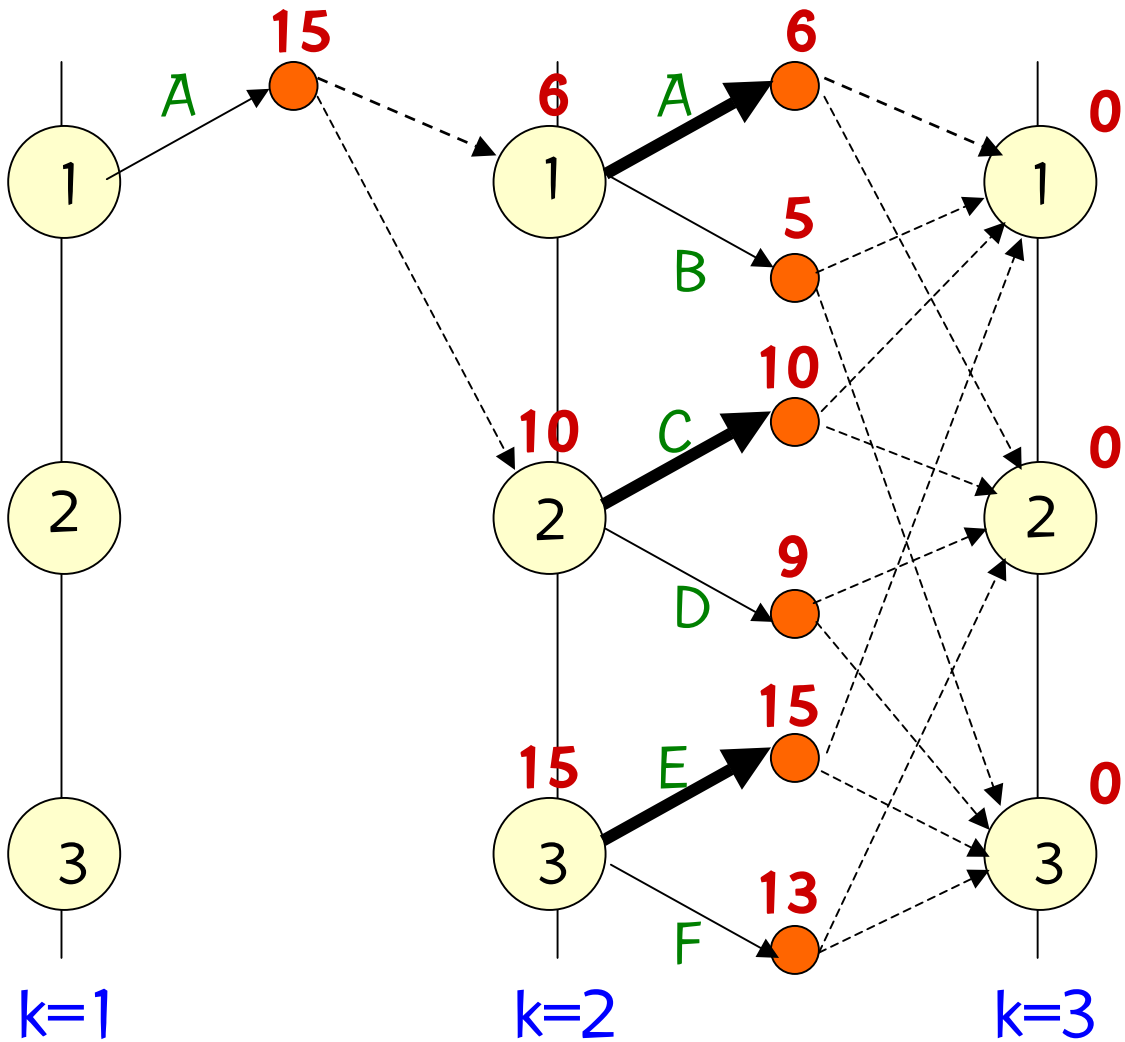
  Nesta etapa, a escolha da roleta B proporciona um maior ganho esperado.

- Os mesmos cálculos podem ser feitos para as regiões 2 e 3.

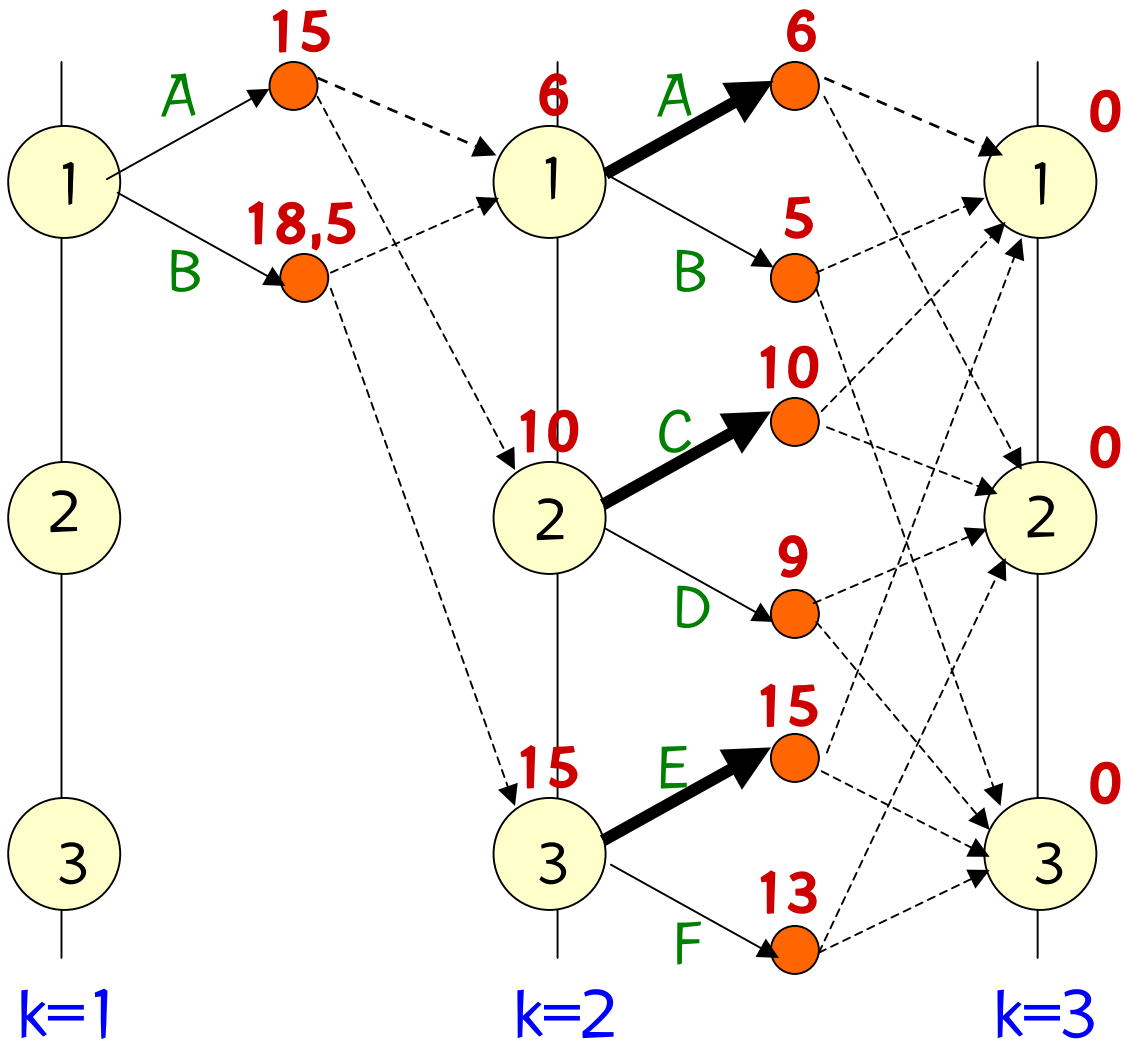
Programação dinâmica estocástica



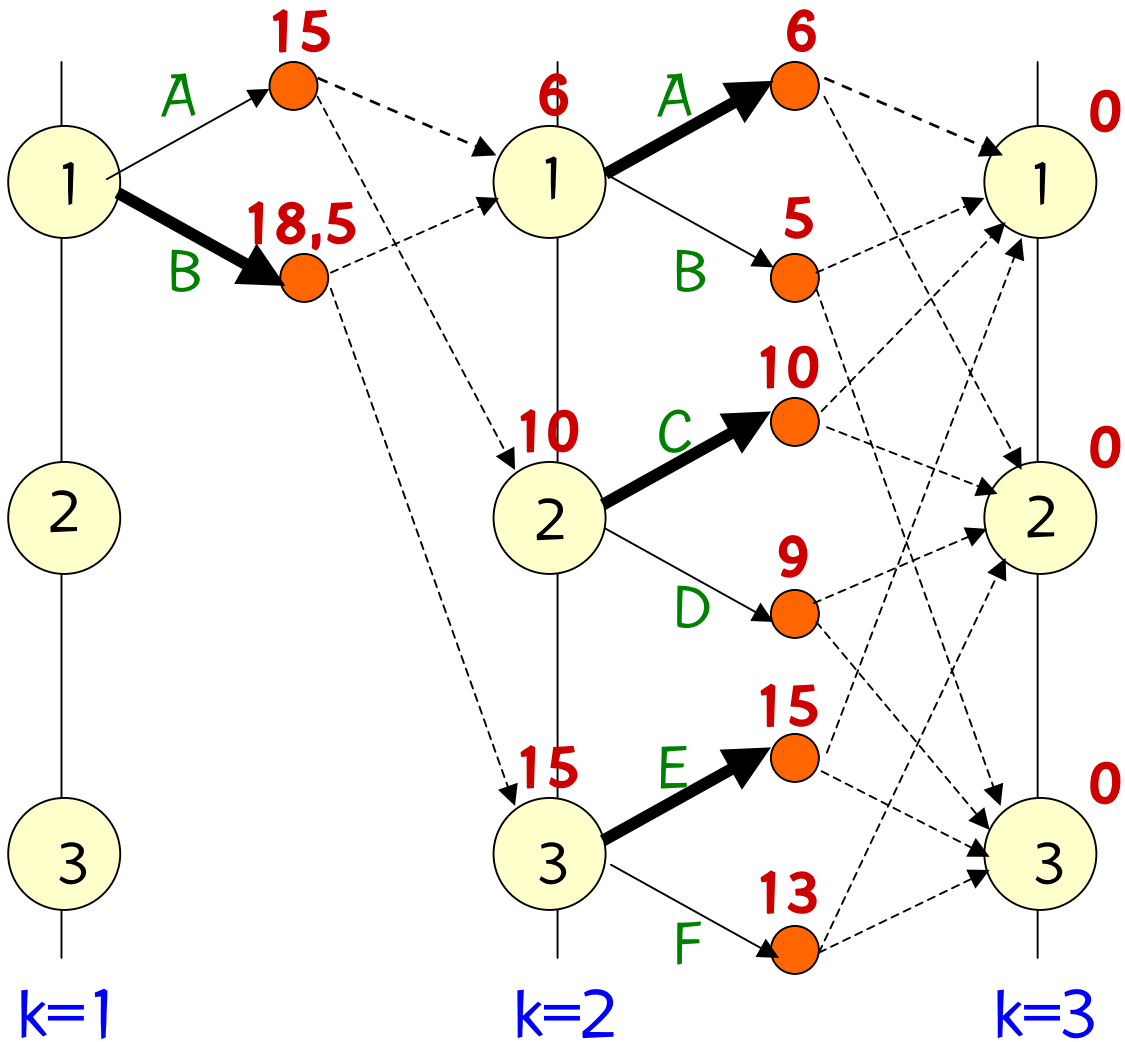
Programação dinâmica estocástica



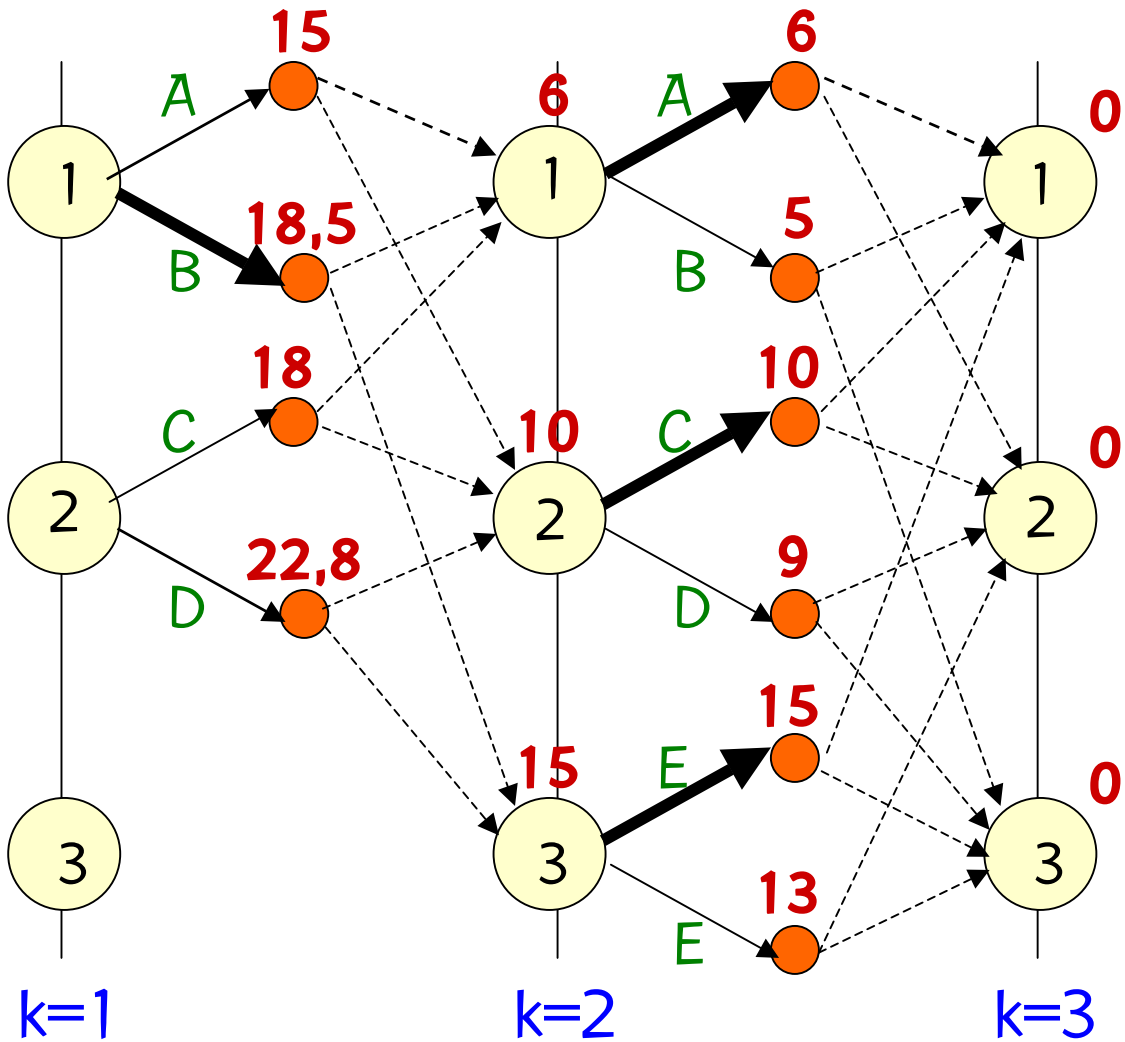
Programação dinâmica estocástica



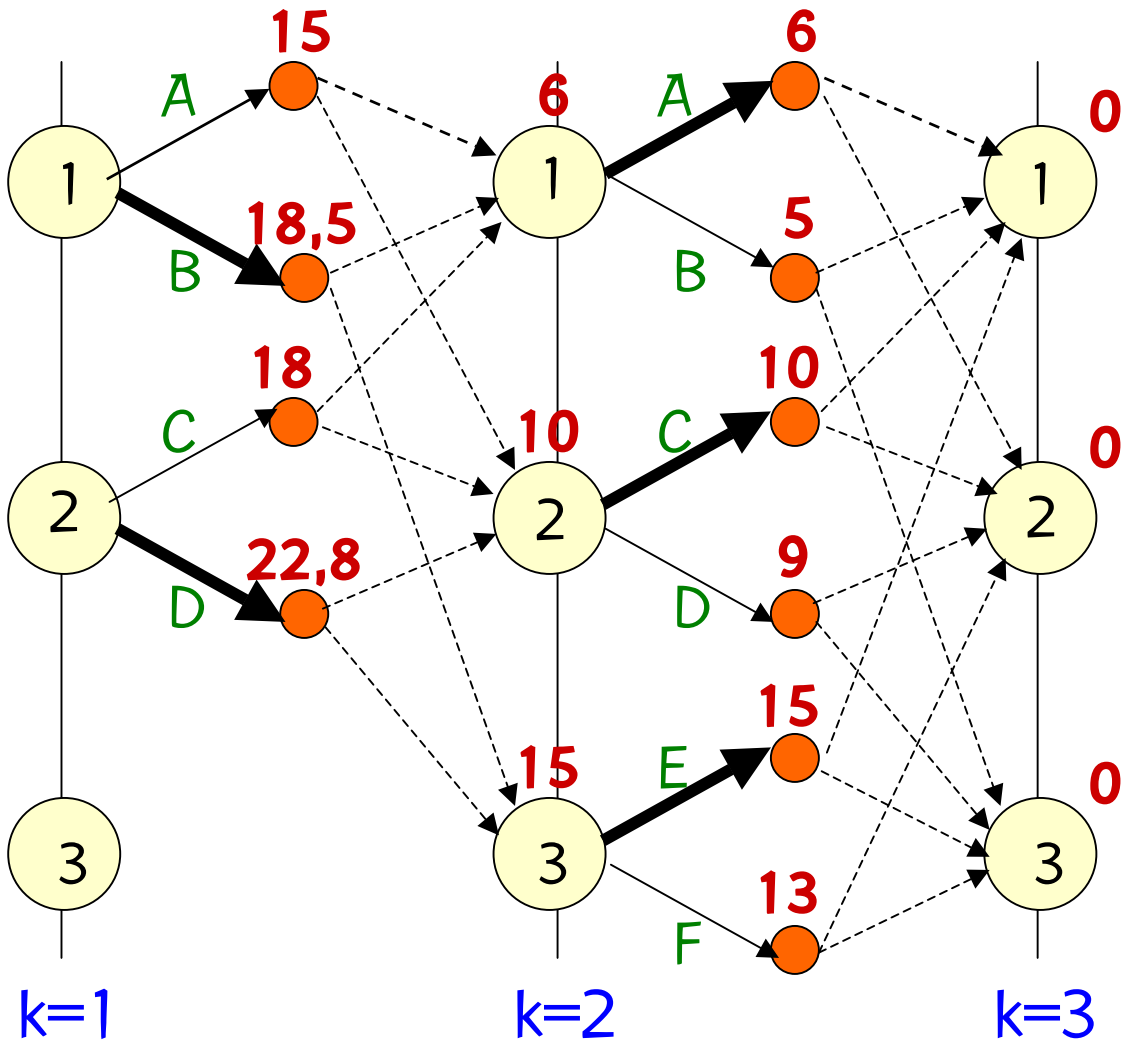
Programação dinâmica estocástica



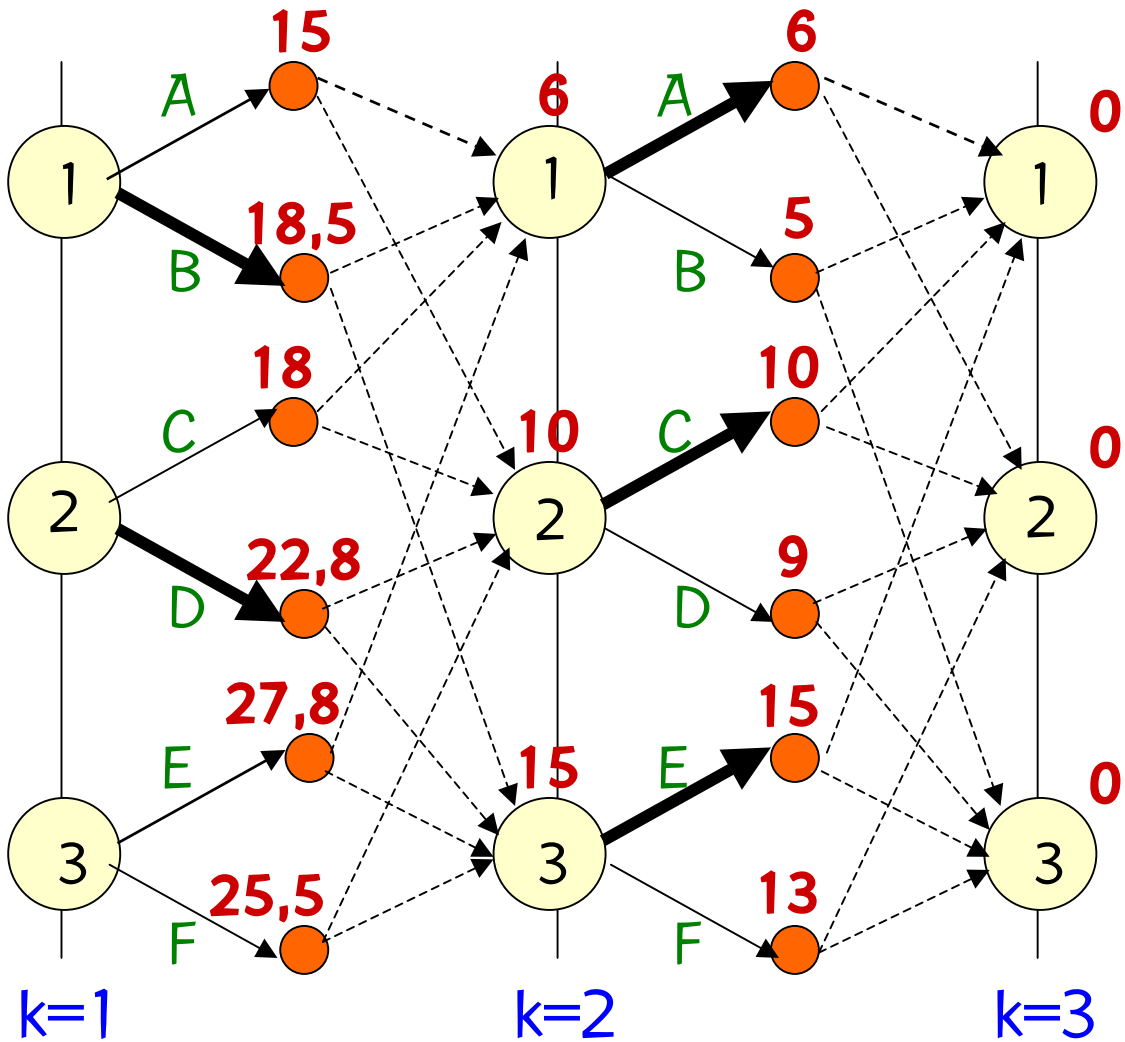
Programação dinâmica estocástica



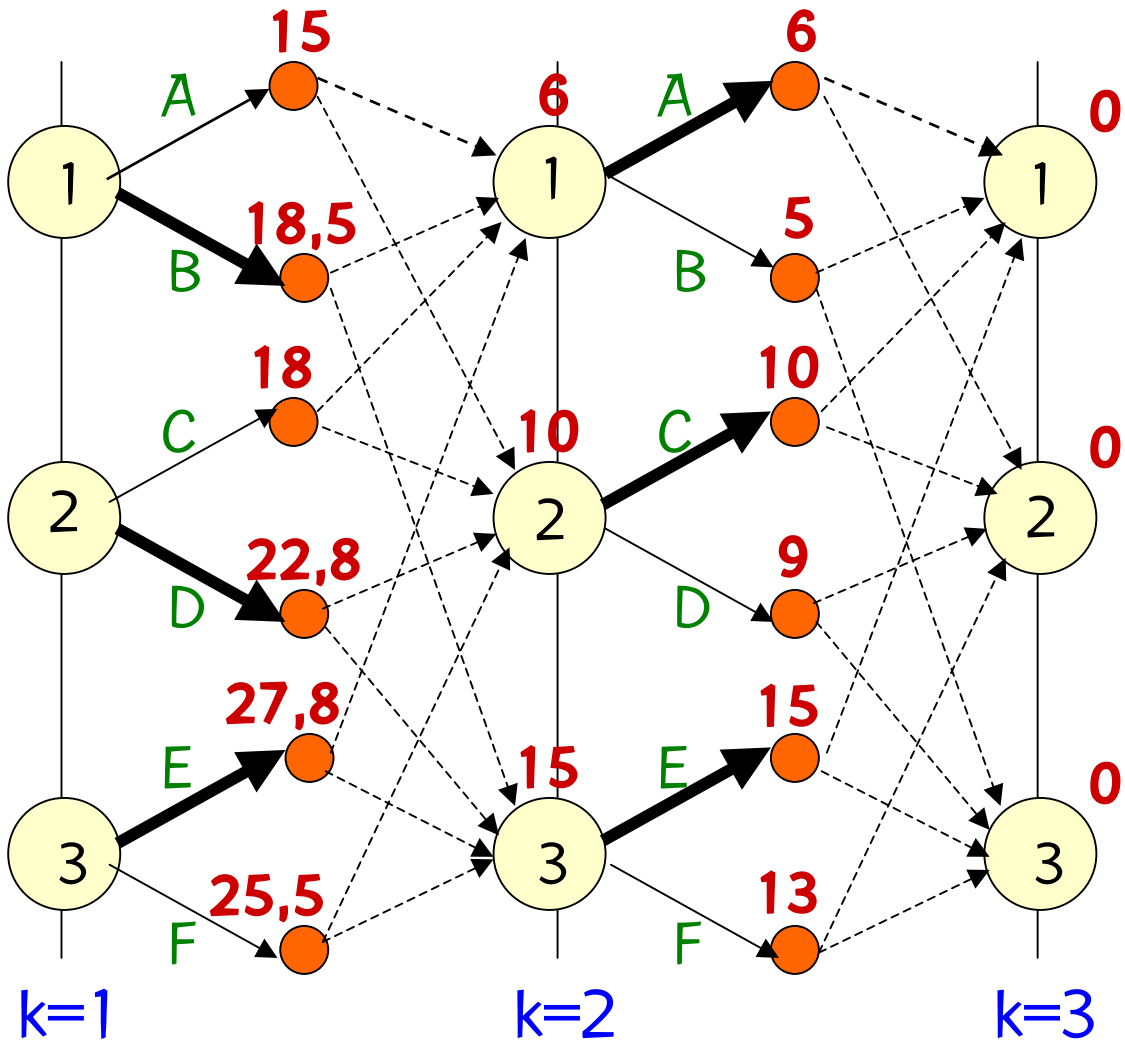
Programação dinâmica estocástica



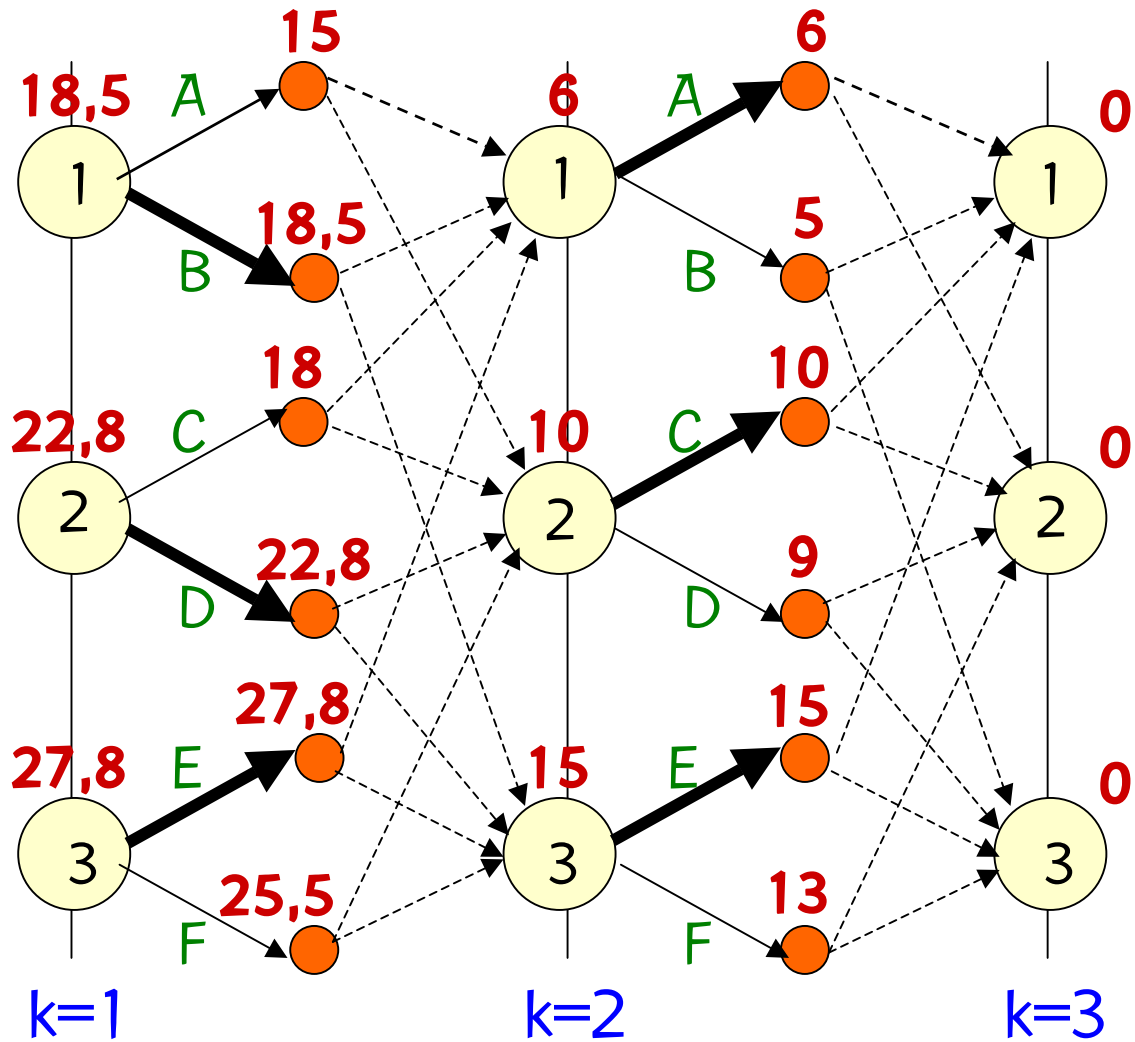
Programação dinâmica estocástica



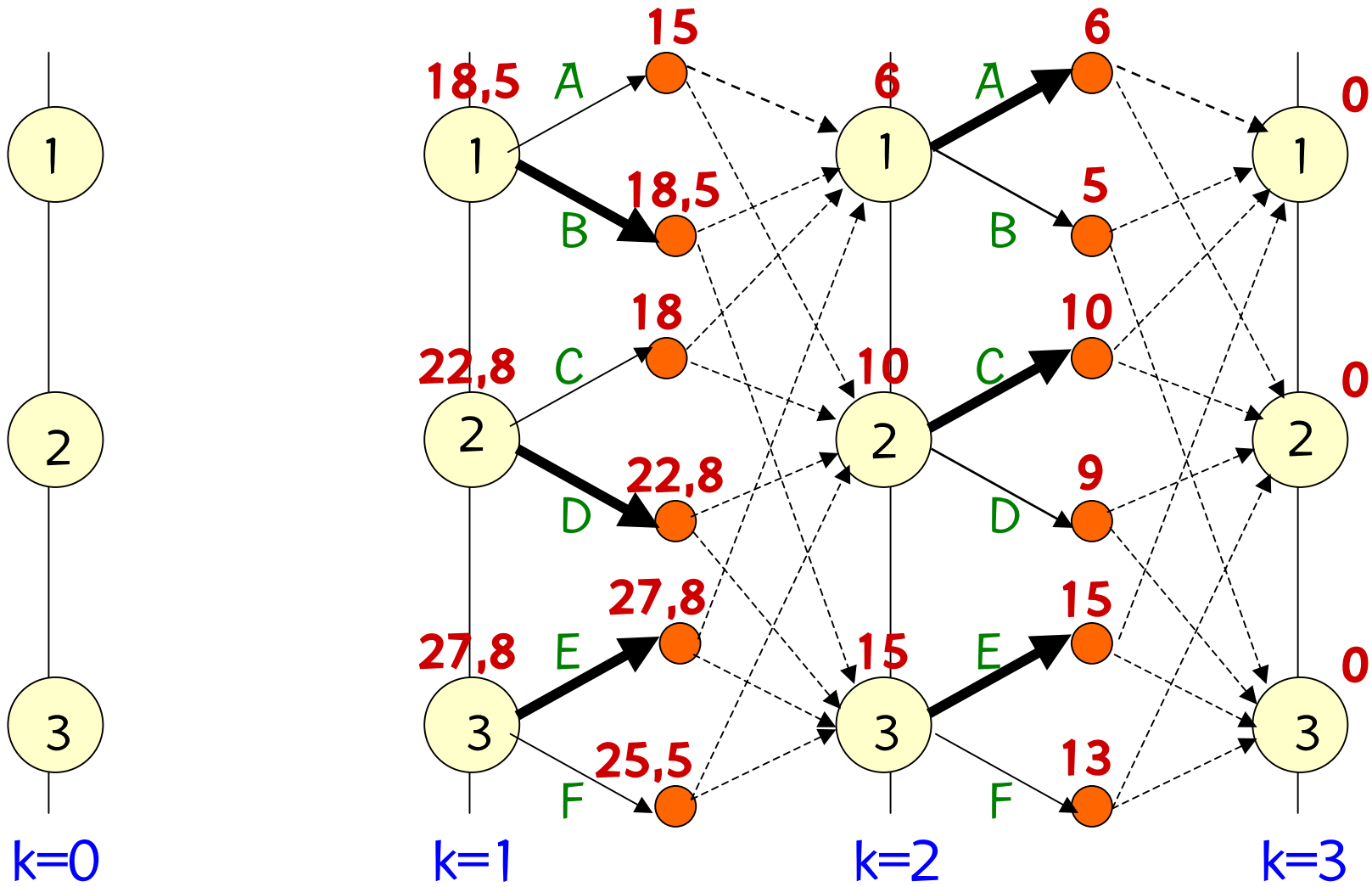
Programação dinâmica estocástica



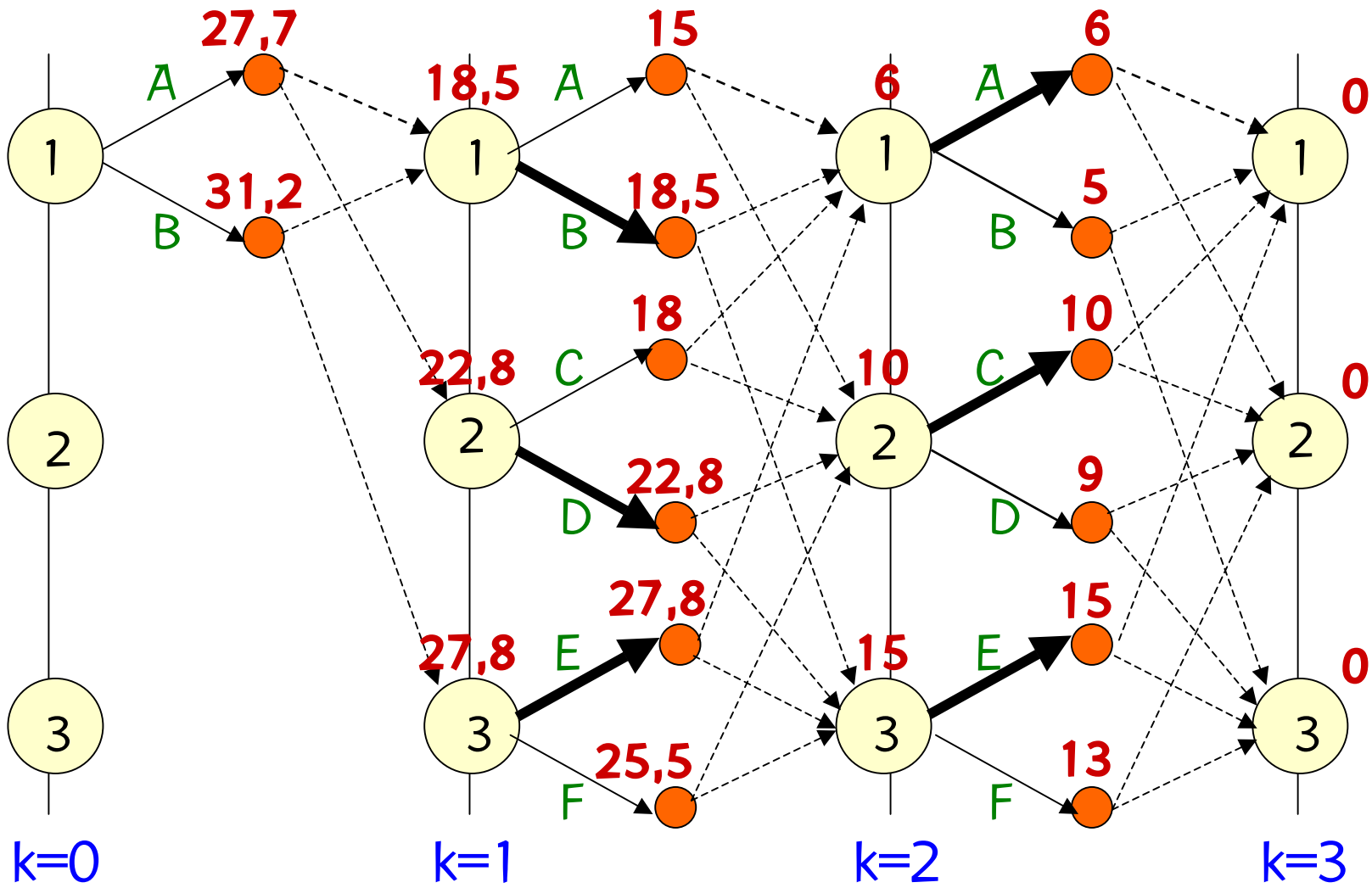
Programação dinâmica estocástica



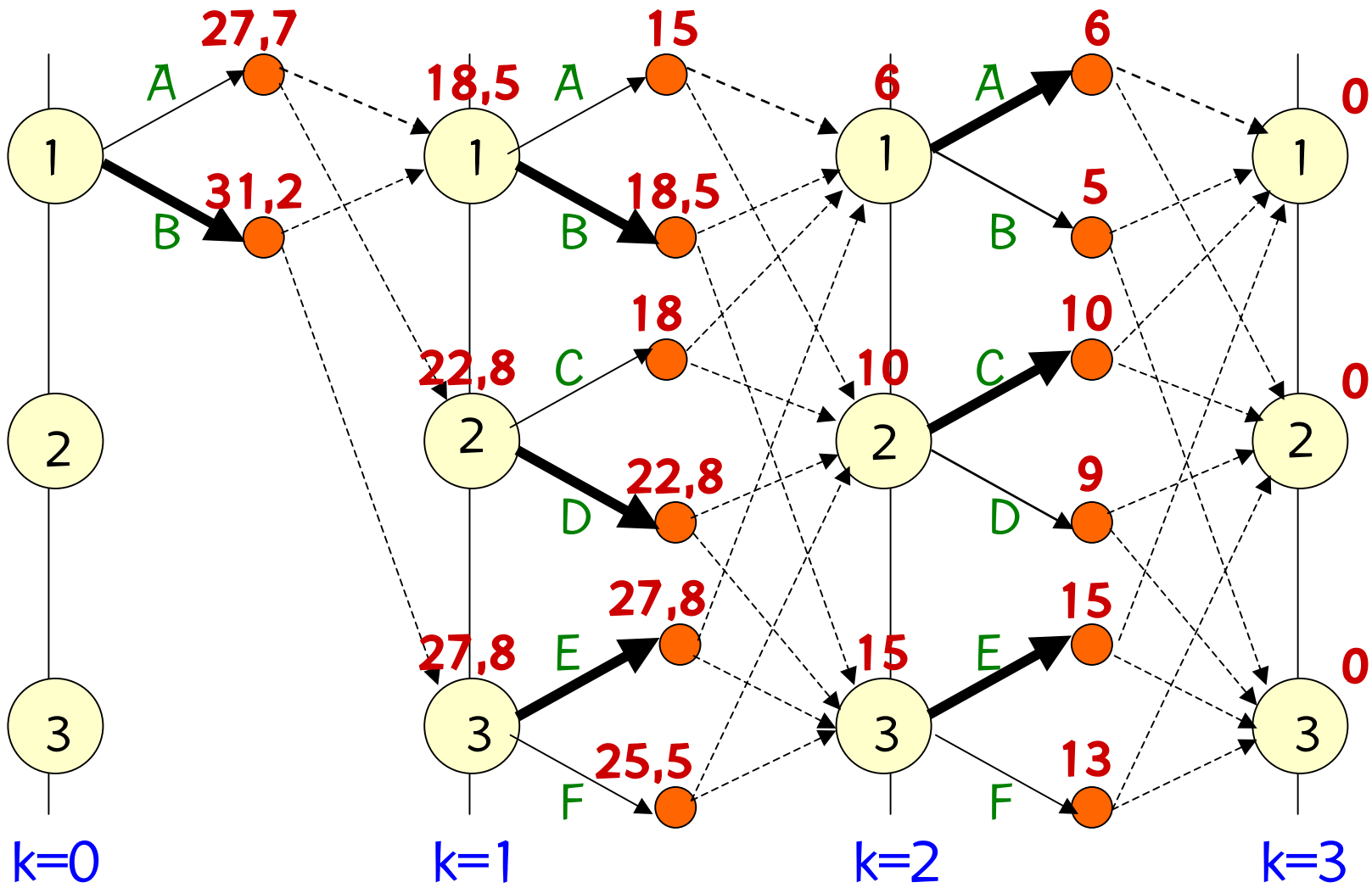
Programação dinâmica estocástica



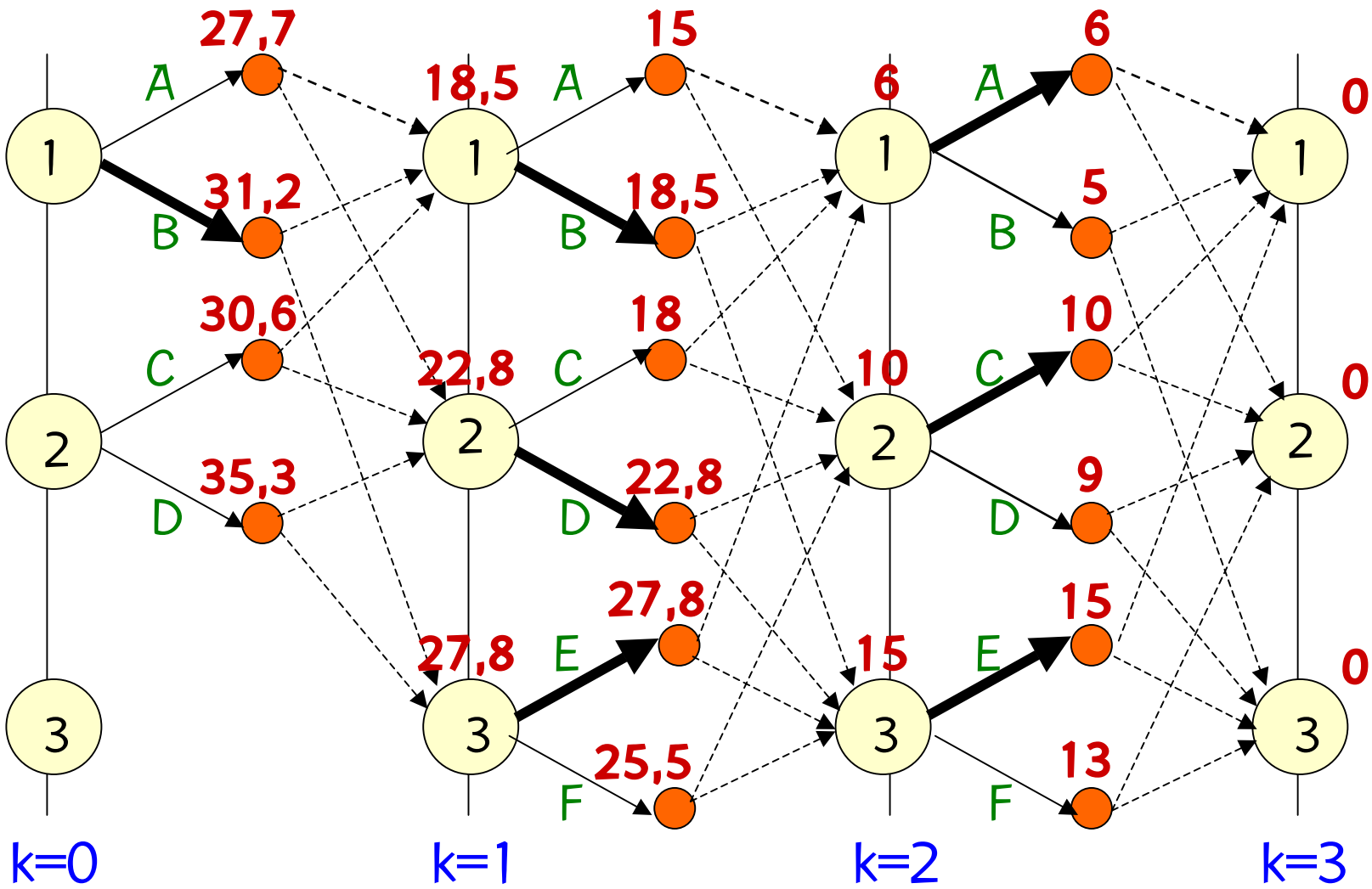
Programação dinâmica estocástica



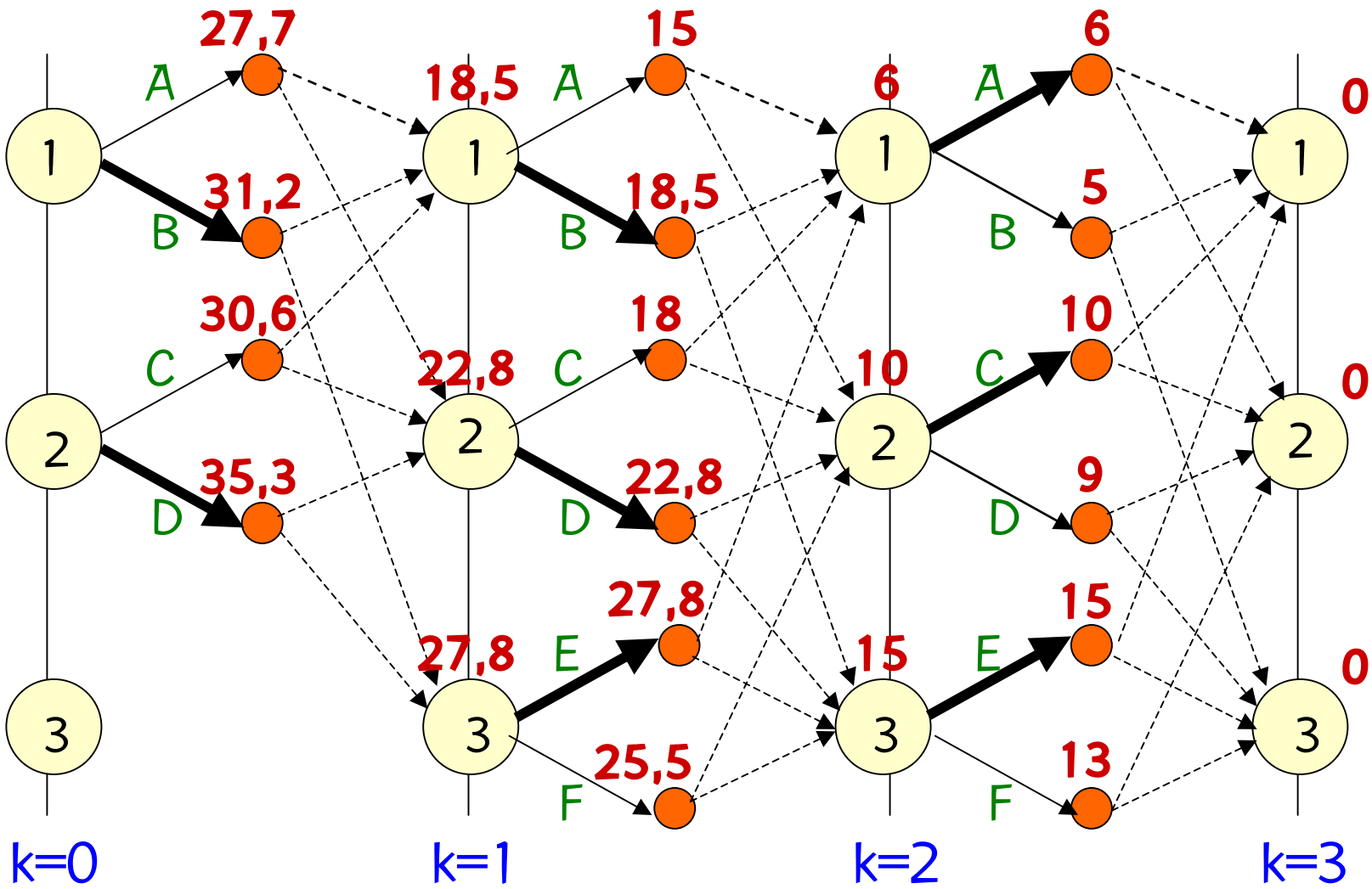
Programação dinâmica estocástica



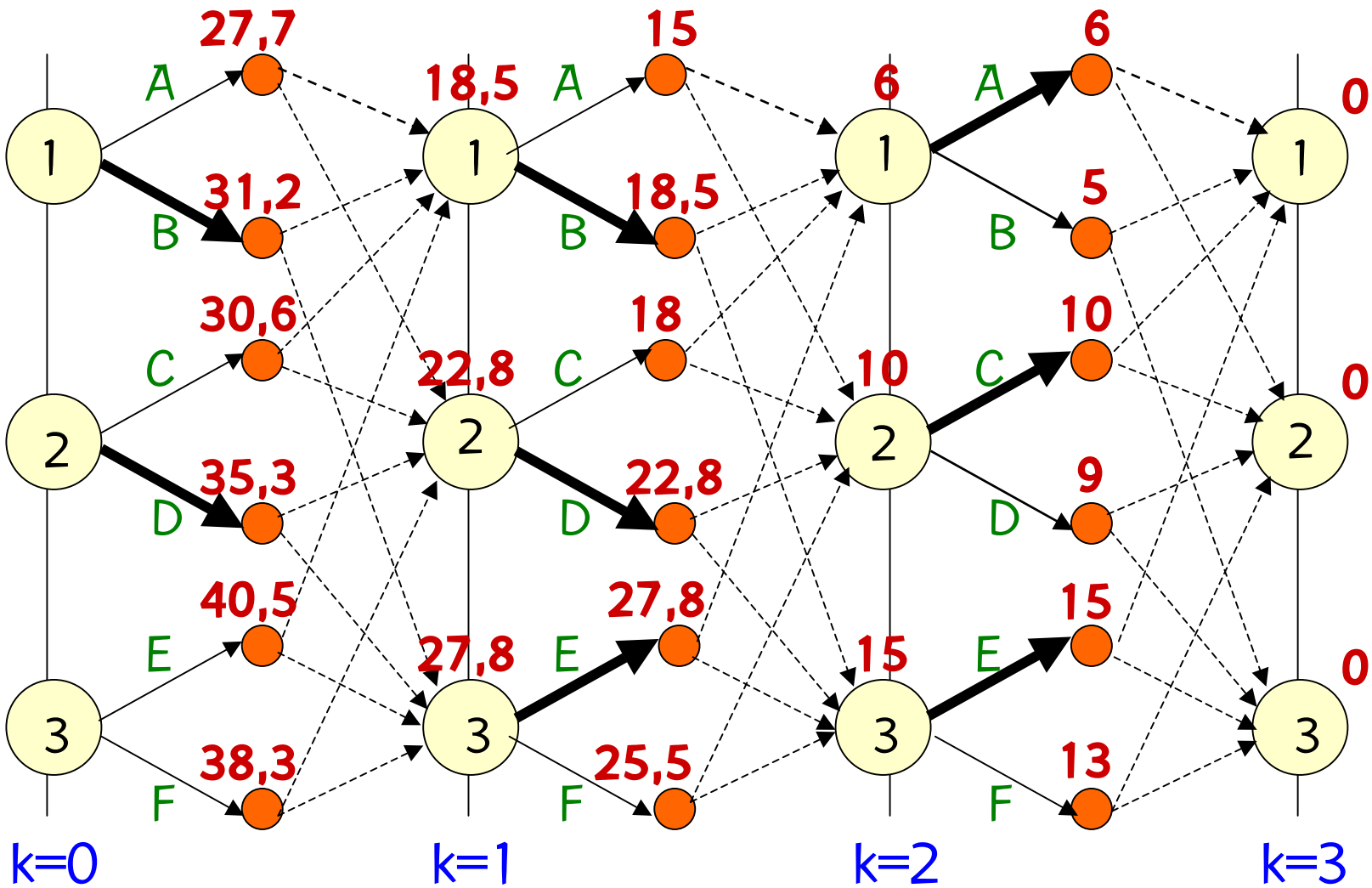
Programação dinâmica estocástica



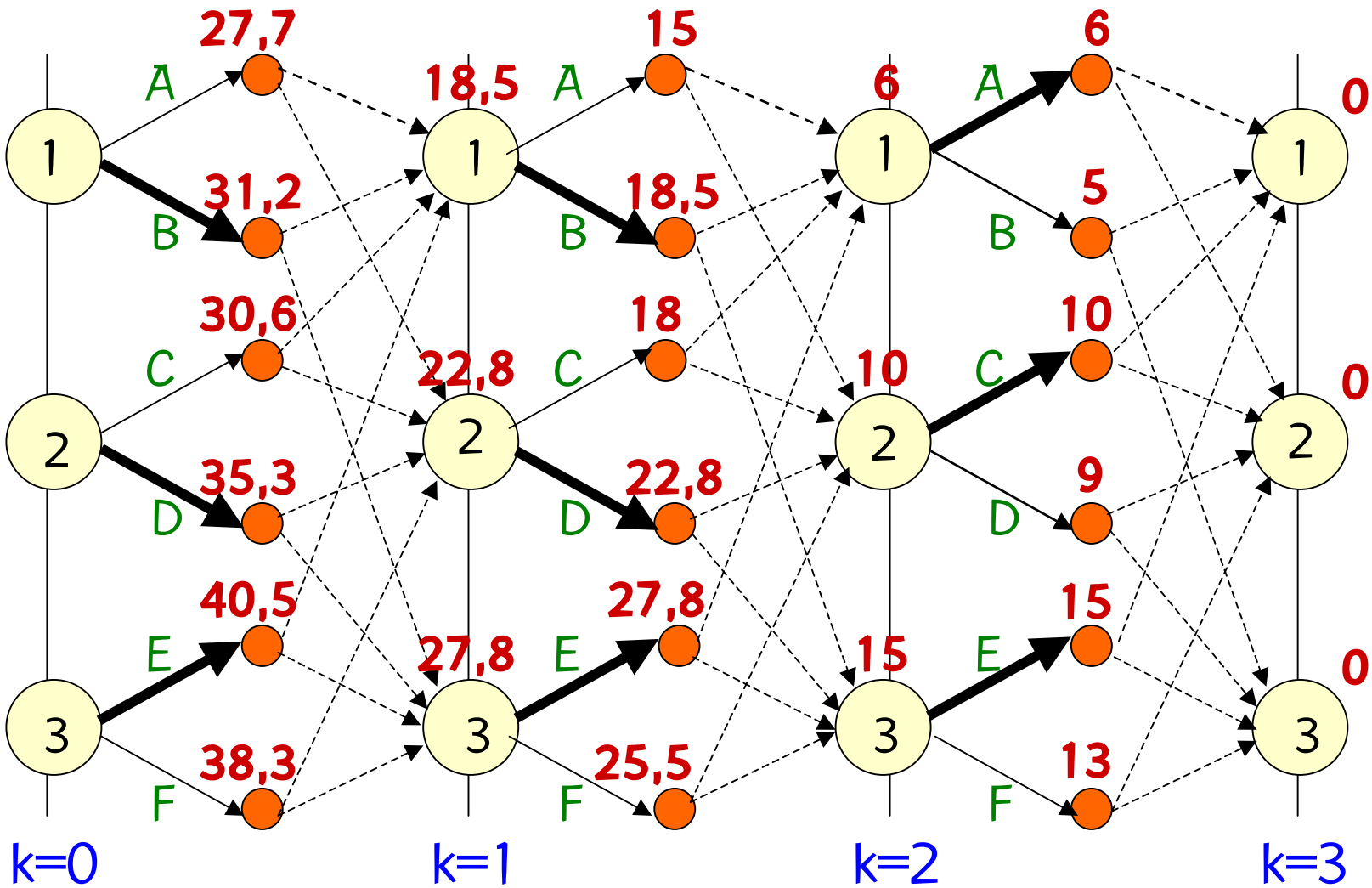
Programação dinâmica estocástica



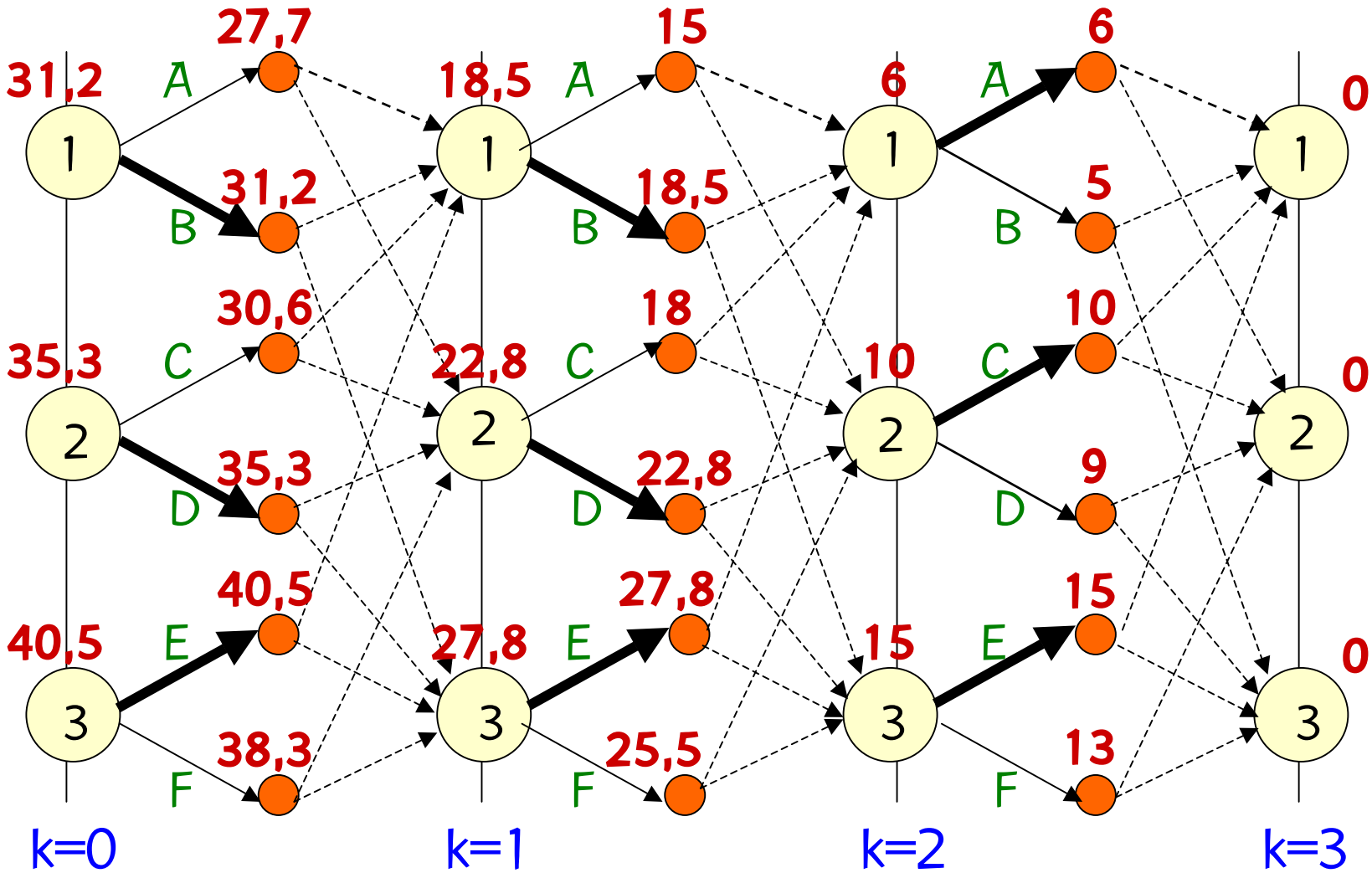
Programação dinâmica estocástica



Programação dinâmica estocástica



Programação dinâmica estocástica



Programação dinâmica estocástica

- Não existe uma trajetória ótima, pois os resultados das transições não são conhecidos.
- A solução é uma estratégia ótima, que informa a decisão ótima a partir de cada estágio de cada estado:

  Estágio k = 0 (jogada 1): 1 - B, 2 - D, 3 - E

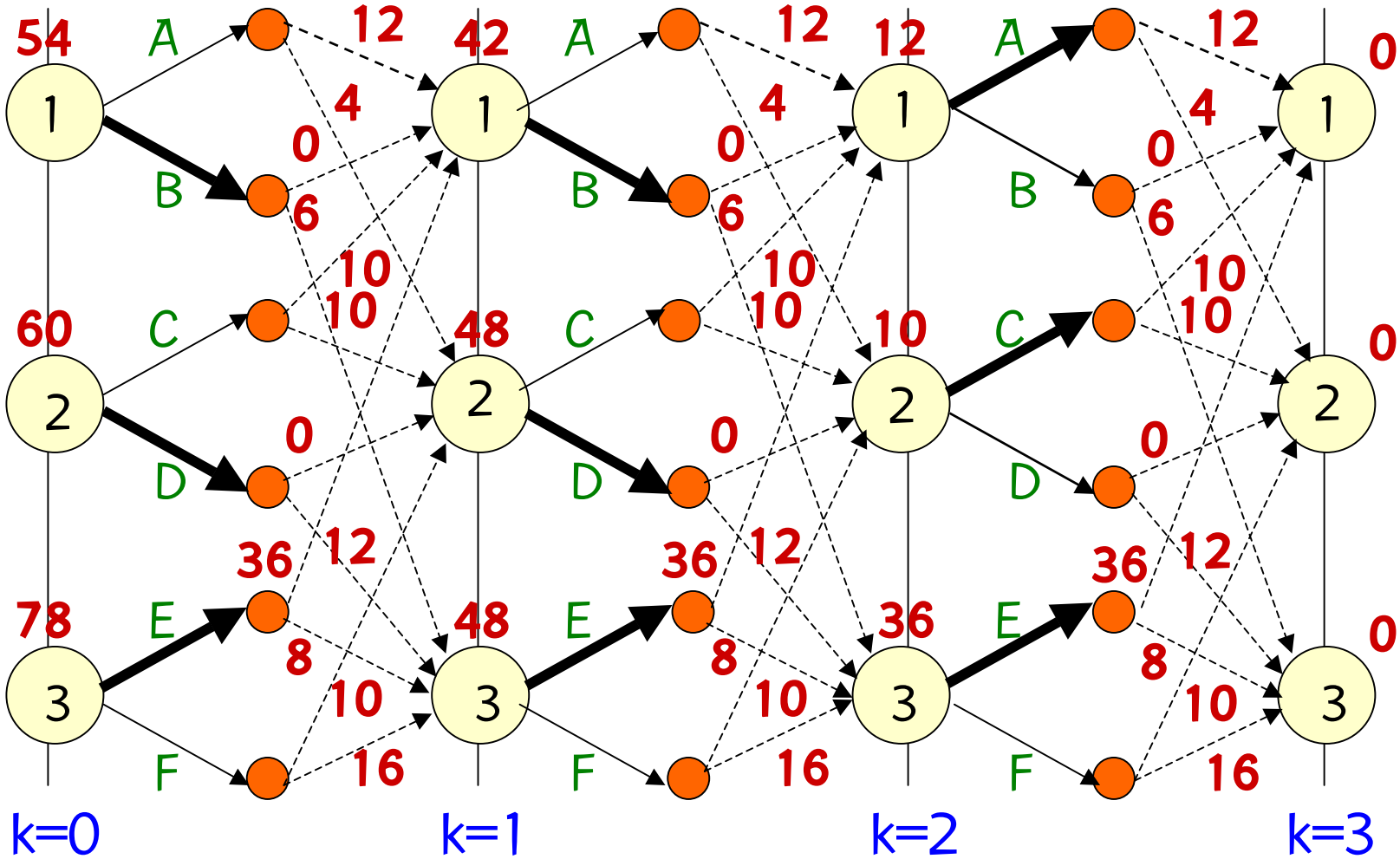
  Estágio k = 1 (jogada 2): 1 - B, 2 - D, 3 - E

  Estágio k = 2 (jogada 3): 1 - A, 2 - C, 3 - E
- Ganho esperado máximo = (1/3) x 31,2 + (1/3) x 35,3 + (1/3) x 40,5 = 35,7

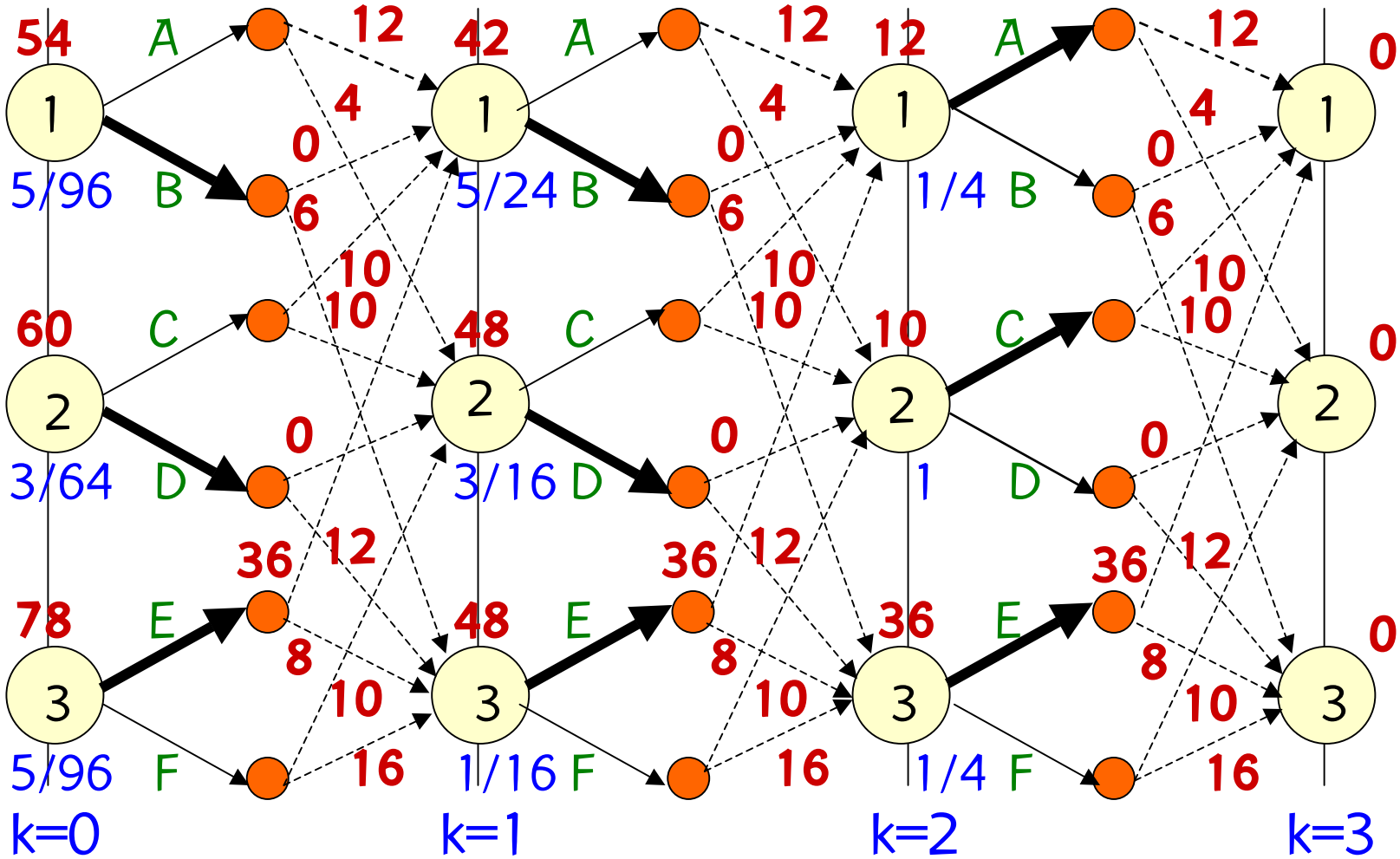
Programação dinâmica estocástica

- Um problema de PD probabilístico pode conter alguns elementos determinísticos, tais como o estado inicial, o estado final ou o resultado de algumas decisões.
- Qual o maior e o menor ganho que podem ser obtidos com a estratégia ótima e quais são suas probabilidades?
- Estas informações podem ser calculadas de forma determinística:
  - ganho máximo = 78, probabilidade = (5/96) x (1/3) = 5/288
  - ganho mínimo = 4, probabilidade = 1/144

Programação dinâmica estocástica



Programação dinâmica estocástica



Programação dinâmica estocástica

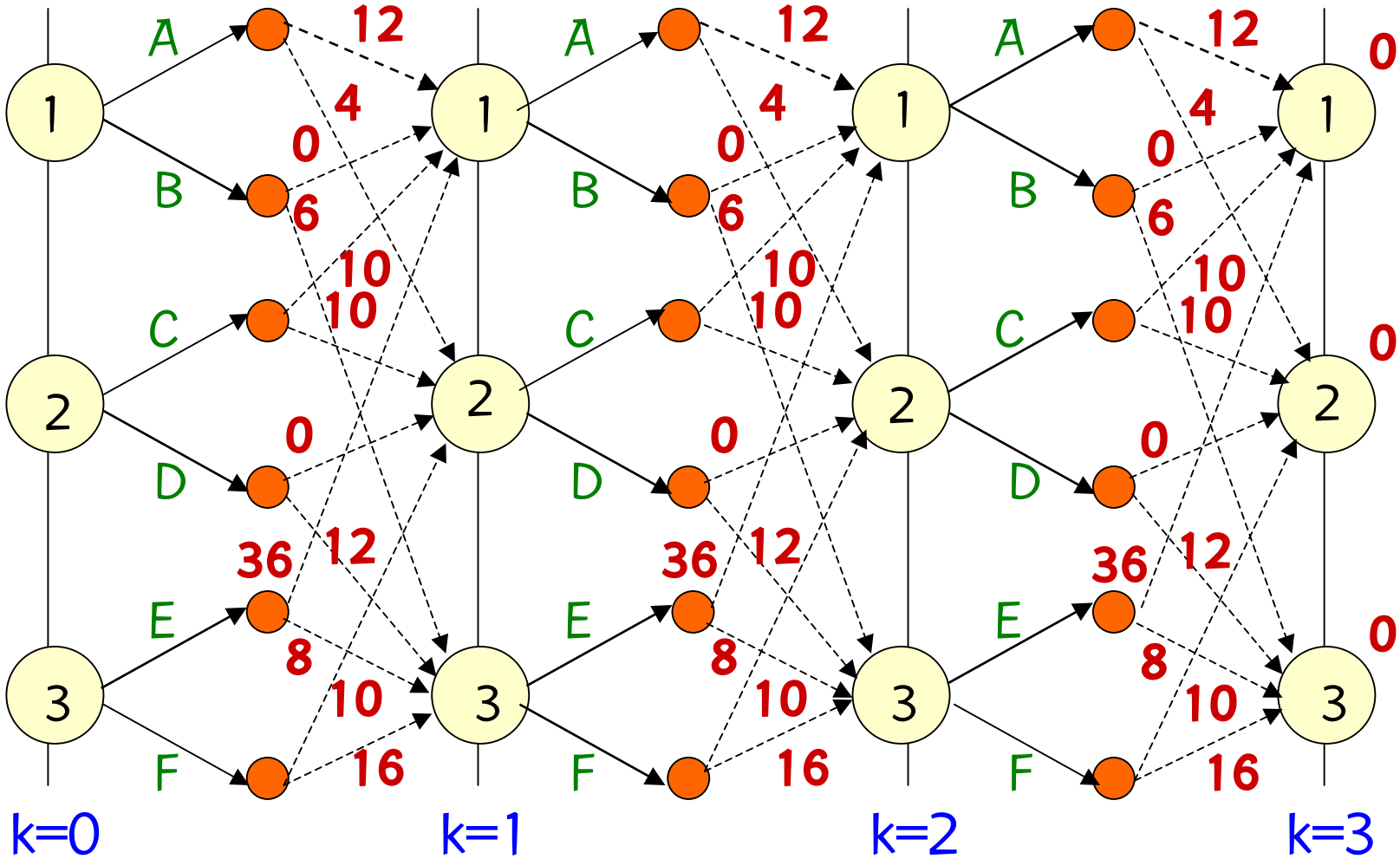
- Como joga um jogador que gosta de arriscar e acredita na sua sorte?
  - Escolher sempre a roleta cujo maior ganho é máximo.

    Estágio k = 0 (jogada 1): 1 - B, 2 - D, 3 - E

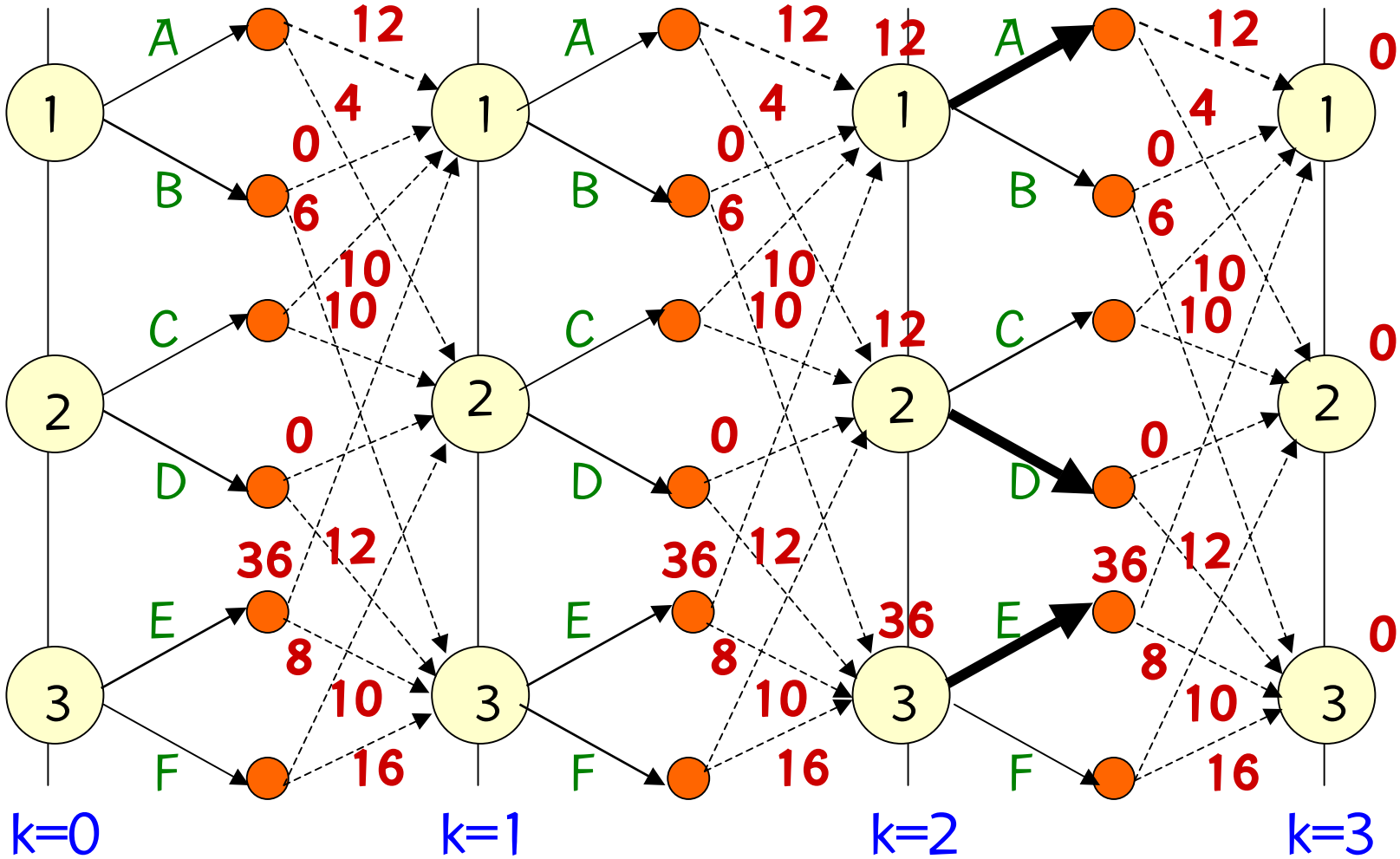
    Estágio k = 1 (jogada 2): 1 - B, 2 - D, 3 - F

    Estágio k = 2 (jogada 3): 1 - A, 2 - D, 3 - E
- Como joga um jogador conservador que não gosta de arriscar, pois não acredita na sua sorte?
  - Escolher sempre a roleta cujo menor ganho é máximo.

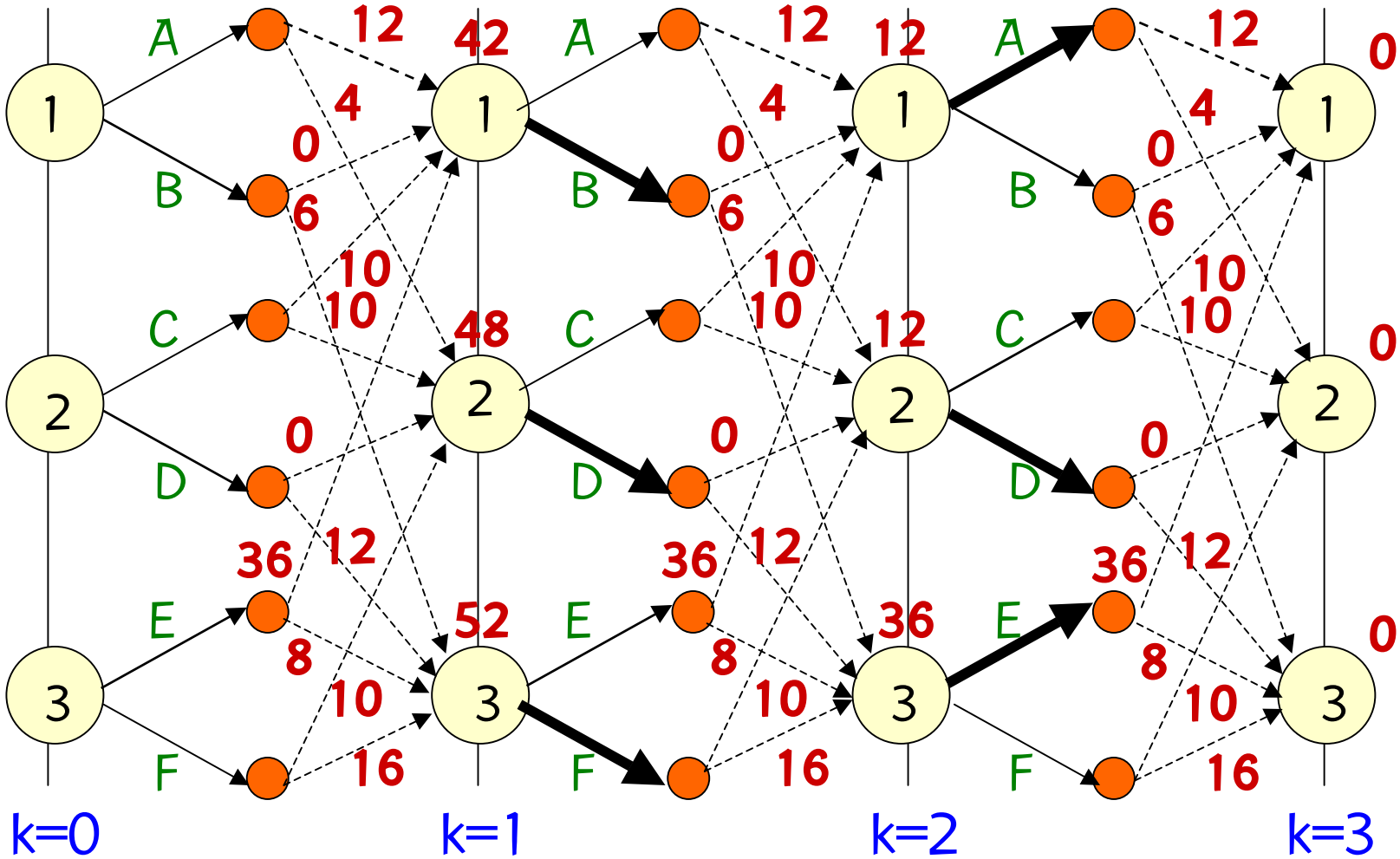
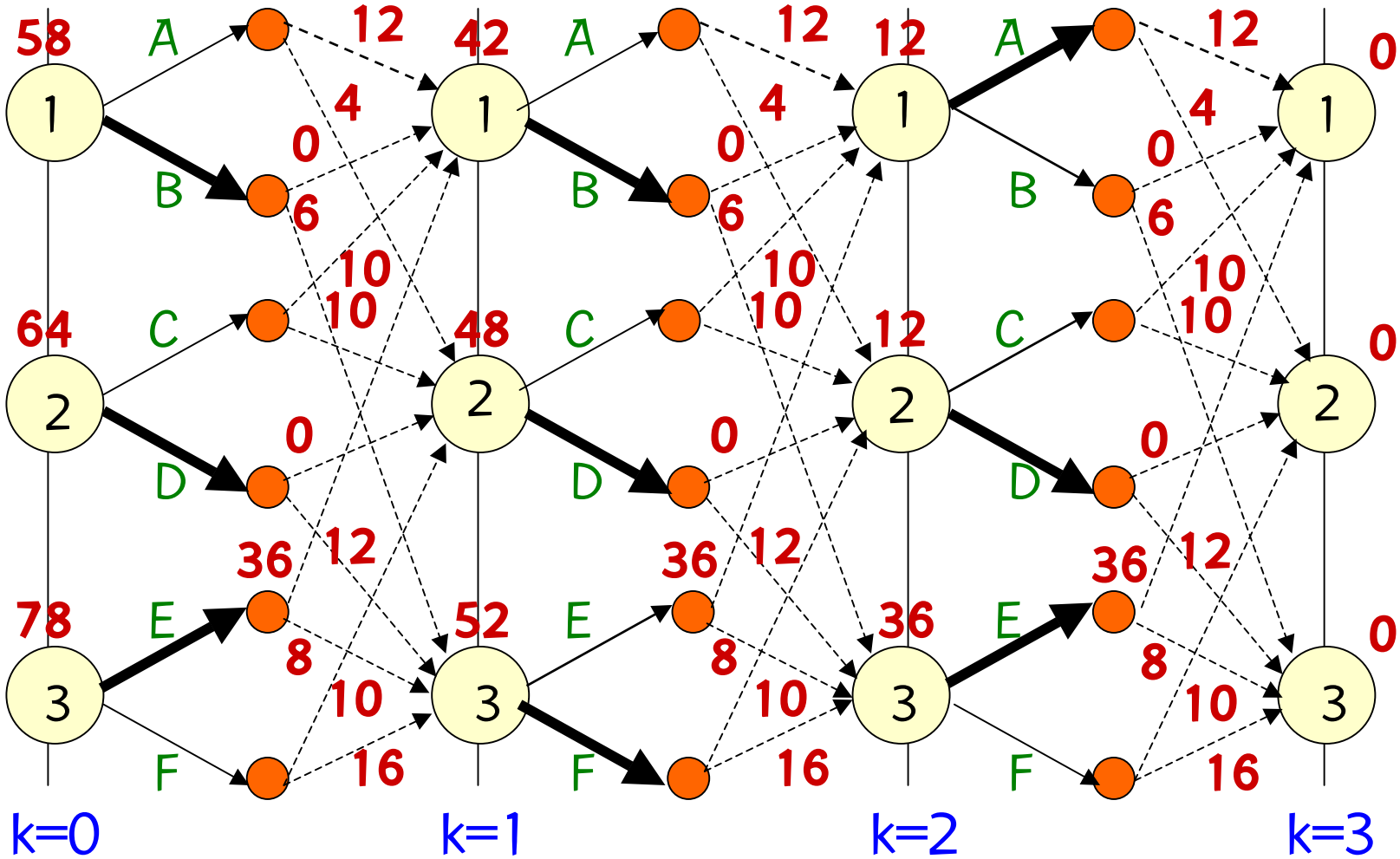
Programação dinâmica estocástica



Programação dinâmica estocástica



Programação dinâmica estocástica



Programação dinâmica estocástica