Dynamic Programming
動態規劃

張智星 (Roger Jang)
jang@mirlab.org
http://mirlab.org/jang
多媒體資訊檢索實驗室
台灣大學 資訊工程系

Dynamic Programming

- Dynamic Programming (DP)
  - An effective method for finding the optimum solution to a multi-stage decision problem, based on the principal of optimality
- Applications: NUMEROUS!

Quiz!

  - Longest common subsequence, edit distance, matrix chain products, all-pair shortest distance, dynamic time warping, hidden Markov models, …

Principal of Optimality

- Richard Bellman, 1952
  - An optimal policy has the property that whatever the initial state and the initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Web Resources about DP

- Recordings on the web
  - MIT Open Course Ware

Problems Solvable by DP

- Characteristics of problems solvable by DP

Quiz!

  - Decomposition: The original problem can be expressed in terms of subproblems.
  - Subproblem optimality: the global optimum value of a subproblem can be defined in terms of optimal subproblems of smaller sizes.

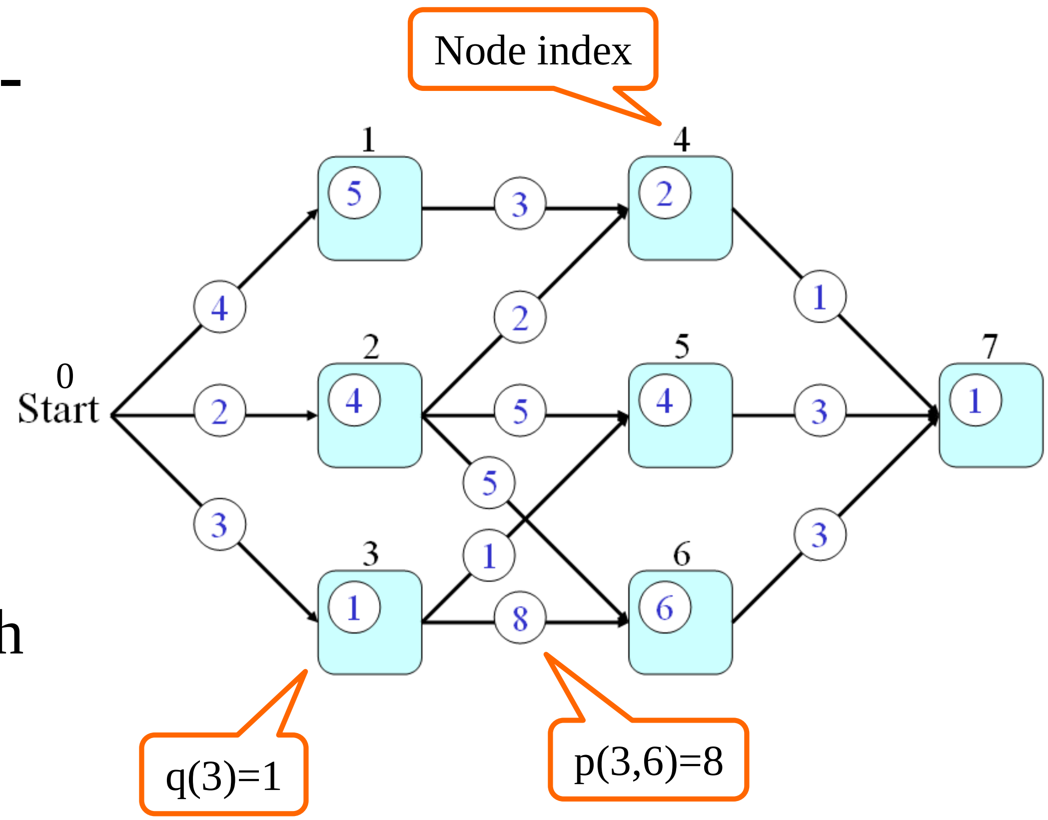
Three-step Formulas of DP

Quiz!

- DP formulation involves 3 steps
  - Define the optimum-value function for recursion
  - Derive the recurrent formula of the optimum-value function, with boundary conditions
  - Specify the answer to the original task in terms of the optimum-value function.

DP Example: Optimal Path Finding

- Path finding in a feed-forward network
  - p(a,b): transition cost
  - q(a): state cost
- Goal
  - Find the optimal path from nodes 0 to 7 such that the total cost is minimized.



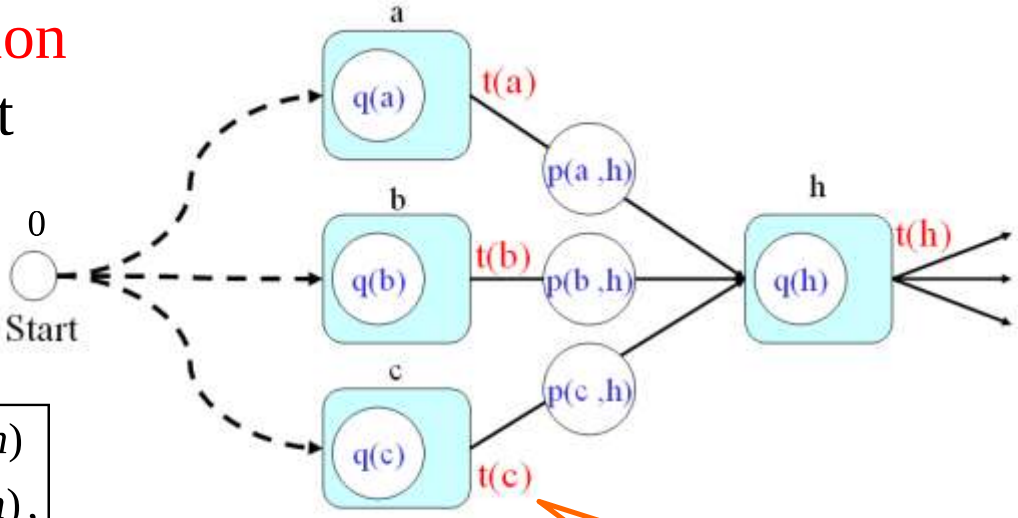
DP Example: Optimal Path Finding

- Three steps in DP
  - Optimum-value function
    t(h): the minimum cost from the start point (node 0) to node h.
  - Recurrent formula

$$t(h) = q(h) + \min \begin{cases} t(a) + p(a,h) \\ t(b) + p(b,h), \\ t(c) + p(c,h) \end{cases}$$

with boundary condition $t(0) = 0$.

  - Answer: t(7)

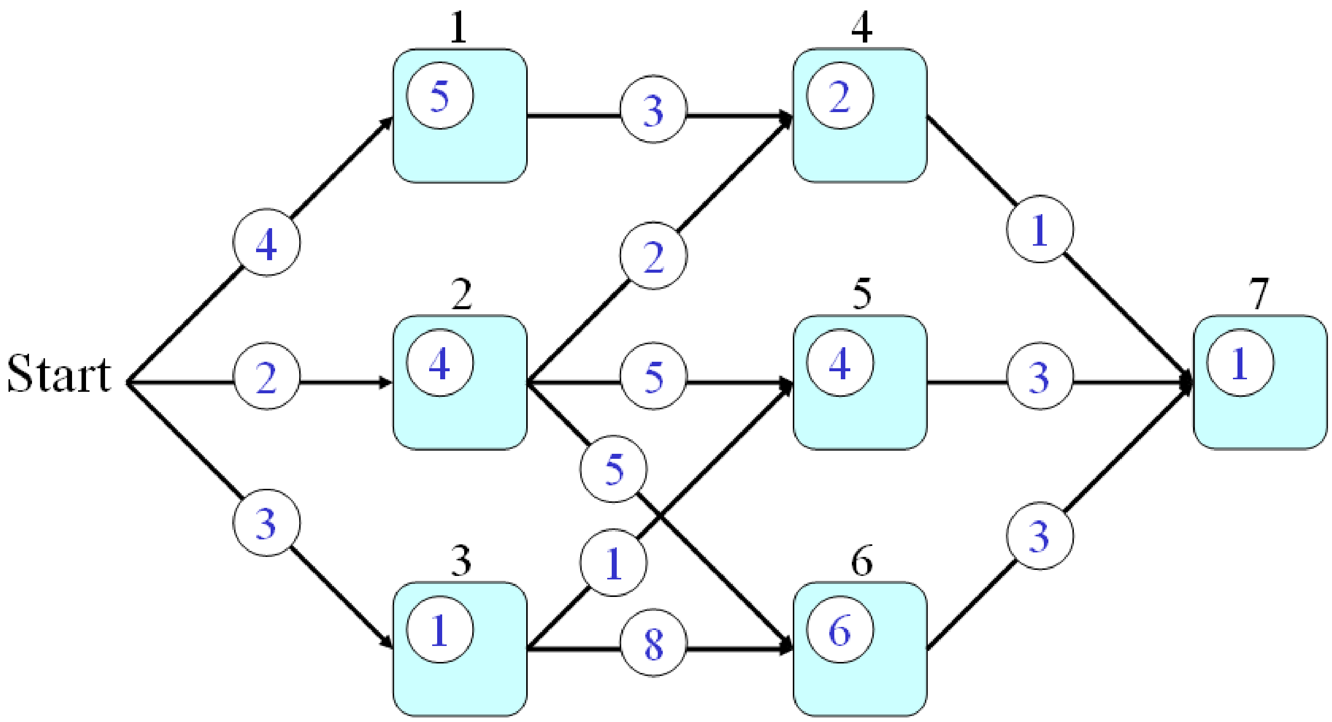


Optimum-value function

Last node

DP Example: Optimal Path Finding

- Walkthrough of DP
  - Flash
  - Gif

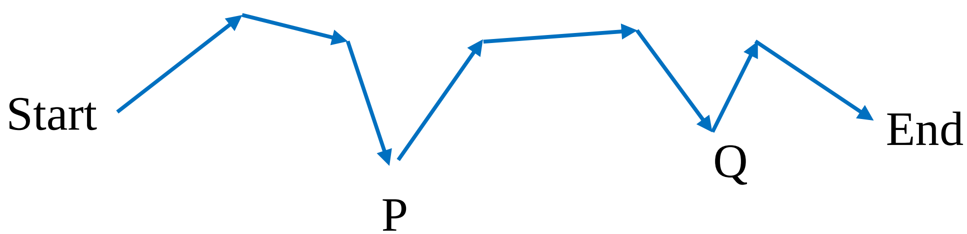


Observations

- Some observations based on this path finding example
  - Once t(7) is found, t(k), ∀k<7 is also found
  - Multi-stage ➔ Layer-by-layer computation ➔ No loops in the graph
- In fact
  - Any DP problem can be visualized as this optimal path finding problem!

Principal of Optimality: Example

- In terms of the min-cost path problem
  - Any sub-path of the min-cost path should itself be a min-cost path given the starting and ending nodes



$P \rightarrow Q$

$\wedge Q \rightarrow \wedge P$

Bottom-up Approach of DP

- Usually bottom-up design of DP
  - Start from the bottom (base cases)
  - Solve small sub-problems
  - Store solutions
  - Reuse previous results for solving larger sub-problems

Usually it's reduced to path finding via table filling!

Characteristics of DP

- Some general characteristics of DP

Quiz!

  - We need to store back-tracking information in order to identify the path efficiently.
  - Once the optimal path is found, all the related sub-problems are also solved.
  - DP can only find the optimal path. To find the second best, we need to resort to a more complicated n-best approach.